ISSN 1870-4069

Reconstrucción autónoma de un mapa métrico de una pista de competencias usando visión artificial

Esther Diaz Sarmientos[1], Humberto Sossa Azuela[2], Alberto Petrilli Barceló[3]

[1] Universidad Tecnológica de la Mixteca, División de Estudios de Posgrado, México

[2] Instituto Politécnico Nacional, Centro de Investigación en Computación, México

[3] Tokyo University of Science, Faculty of Science and Technology, Japan

dise900301@hotmail.com,hsossa@cic.ipn.mx, petrilli@rs.tus.ac.jp

Resumen. El presente artículo describe el desarrollo de un algoritmo para la reconstrucción de un mapa métrico a partir de las imágenes capturadas por la cámara Intel Real Sense a bordo del robot AutoNOMOS mini. Las etapas que componen el algoritmo constan de la corrección de perspectiva de las imágenes, detección de líneas de carril, asociación de puntos clave y la reconstrucción del mapa. Se presentan los resultados obtenidos en una pista real, realizando el procesamiento a bordo y completamente en la plataforma, y de un ambiente simulado en Gazebo en un ambiente virtual.

Palabras clave: Mapeo, Visión Artificial, Gazebo, ROS, autoNOMOS, Proyectividad.

Autonomous Reconstruction of a Metric Map of a Competition Track Using Computer Vision

Abstract. This article describes the development of an algorithm for the reconstruction of a metric map from the images captured by the Intel Real Sense camera on board the AutoNOMOS mini robot. The stages that make up the algorithm consist of the perspective correction of the images, detection of lane lines, association of key points and the reconstruction of the map. The results obtained are presented on a real track, carrying out the processing on board and

completely on the platform, and from a simulated environment in Gazebo in a virtual environment.

Keywords: Mapping, Artificial Vision, Gazebo, ROS, autoNOMOS, Projectivity.

1. Introducción

En robótica los mapas constituyen una herramienta importante en la representación del entorno y permiten planeación de rutas, localización y navegación. Generalmente se dividen en topológicos y métricos [1]. En esta clasificación, los de tipo métrico se distinguen por ser menos complejos en cuanto a la representación de la información espacial con formas básicas como líneas y curvas.

Las representaciones del entorno dependen completamente de la información proporcionada por los sensores, presentes tanto en el robot como en el ambiente. Estos sensores pueden ser de distintos tipos como láseres, cámaras, GPS, posición y orientación, entre otros.

Los mapas a nivel de carreteras o pistas de competencias contienen información relevante sobre el camino, señales u obstáculos presentes en el entorno. Estas representaciones permiten la planeación de mejores rutas y ahorro de recursos, entre otros beneficios. En las pistas de competencias, contar con información gráfica de carriles permite realizar el reto de recorrer cualquier circuito planeando la trayectoria con anticipación.

En este trabajo se describe la metodología utilizada por medio de visión y asociación de datos para la reconstrucción gráfica de una pista de competencias a nivel de líneas de carril, procesando en tiempo real y a bordo de un vehículo a escala.

2. Trabajos relacionados

La construcción de mapas está basada en la información adquirida por medio de sensores y según las características de relevancia es el tipo de sensor a utilizar. En los coches autónomos, las cámaras permiten la adquisición de información de tipo visual, la cual es base para la detección de líneas de carril. Para esta aplicación por medio de imágenes se usan métodos como redes neuronales y filtros estadísticos [2], [3], [4], [5] y [6].

En el estado del arte de mapas a nivel de carreteras se utilizan distintos sensores a bordo de los vehículos como GPS [7], cámaras [8], sensores láser LiDAR [9], entre otros. El trabajo de [1] realiza una revisión en trabajos en temas de coches autónomos. Dentro de estos destacan las aplicaciones realizadas en mapas tanto métricos como topológicos.

Los vehículos a escala constituyen una oportunidad importante en el desarrollo de aplicaciones para coches autónomos. En este ámbito destacan las competencias internacionales como la copa Carolo [10] de la Technische Universität Braunschweig en Alemania, como nacionales con el Torneo Mexicano de Robótica [11] o Talent Land [12] por la oportunidad que ofrecen de desarrollar algoritmos para resolver los retos.

Fig. 1. AutoNOMOS mini V2.

En el trabajo de [13] hacen un recuento de los algoritmos utilizados por cada equipo en la competencia del 2016 para resolver los distintos desafíos de los que consta la copa Carolo. [14] desarrolló una metodología para la reconstrucción de una pista de competencias utilizando un mapa métrico basado en ocupación de rejillas. Su sistema realiza la construcción de un mapa local por medio de la estimación de movimiento y la extracción de características de imágenes. Los mapas locales son agregados a un mapa global por medio de un algoritmo de asociación de datos.

En el ámbito nacional, [15] desarrolló un sistema de navegación para la plataforma AutoNOMOS mini V1, donde realiza detección de carriles, generación y seguimiento de rutas, detección y evasión de obstáculos y estacionamiento en paralelo. [16] diseñó una metodología basada en redes neuronales convolucionales para el control de dirección de la plataforma AutoNOMOS mini V1 en ambientes reales y virtuales.

3. Desarrollo

La implementación de este trabajo se realizó a bordo de la plataforma "AutoNOMOS Mini", mostrada en la figura 1. Esta plataforma [17] es un modelo de vehículo (escala 1:10) desarrollado para fines educativos por la Freie Universität Berlín en el entorno de trabajo ROS (Robot Operating System) y cuenta con distintos sensores, entre ellos una cámara Intel Real Sense y un IMU.

ROS es un entorno de trabajo que provee librerías y herramientas para ayudar a desarrollar aplicaciones para robots [18]. Dentro de sus prestaciones importantes está la simulación de robots y entornos de forma gráfica utilizando Gazebo. Gran cantidad de estos modelos se ponen a disposición del público en repositorios en línea. La implementación de este trabajo en un ambiente virtual se realizó con el simulador diseñado por [19]. La metodología consta de cuatro etapas principales, desde la corrección de perspectiva hasta la reconstrucción del mapa. Cada una de estas se describe en las siguientes secciones.

3.1. Calibración de cámaras y corrección de perspectiva

Debido al efecto de la perspectiva en las imágenes obtenidas con la plataforma es necesario realizar una corrección de perspectiva o transformación proyectiva. Una transformación proyectiva o proyectividad [20] es una transformación invertible dada por $H = \mathbb{P}^2 \longrightarrow \mathbb{P}^2$ de manera tal que una linea recta es transformada a una linea recta. La corrección de perspectiva realizada está basada en transformaciones geométricas y

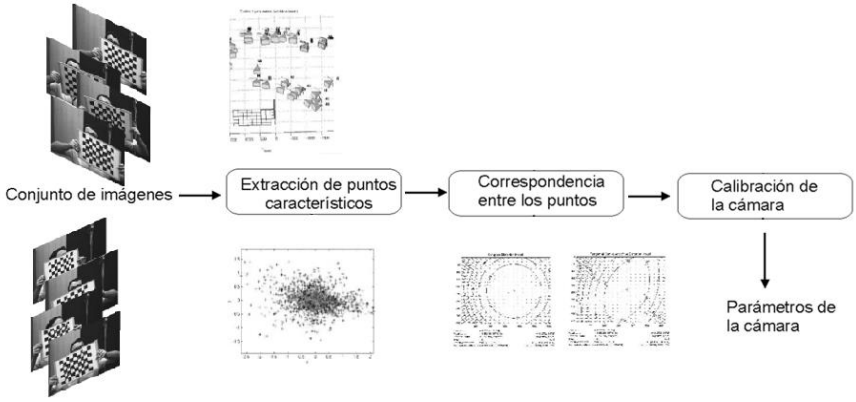


Fig. 2. Proceso de calibración de cámaras [21].

depende de parámetros propios de la cámara obtenidos de la calibración y ubicación de la misma en la plataforma AutoNOMOS mini.

La calibración de una cámara [22] es el proceso que permite obtener los parámetros que definen las condiciones de formación de una imagen, incluyendo la geometría interna y la óptica de la cámara (parámetros intrínsecos), así como su posición y orientación respecto a un patrón de calibración (parámetros extrínsecos). En general el proceso de calibración de cámaras consta de las etapas mostradas en la figura 2.

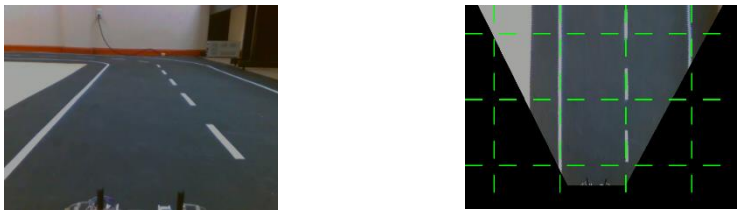
A fin de encontrar los parámetros intrínsecos se realizó la calibración de la cámara Intel Real Sense con un patrón tipo tablero de ajedrez, con un nodo propio de ROS y OpenCV basado en el método de Zhang. En el caso de los parámetros extrínsecos, se definieron dos, la orientación del campo de visión y la distancia del centro óptico respecto al suelo.

Los parámetros usados en la corrección de perspectiva corresponden a una rotación de 286º y una distancia de 298 mm en la plataforma AutoNOMOS mini. En el ambiente virtual se propuso una cámara ideal en cuanto a los parámetros extrínsecos y los valores intrínsecos se ajustaron a una rotación en X de 273º, una traslación en Z de 325 y en Y de 20. Las matrices de transformación obtenidas tanto en el robot real, como en el ambiente virtual están denotadas por las expresiones 1 y 2 respectivamente:

$$H_{real} = \begin{bmatrix} 594.651 & -294.279 & -18985.060 \\ 0 & -71.718 & 90245.466 \\ 0 & -0.961 & 532.637 \end{bmatrix}, \tag{1}$$

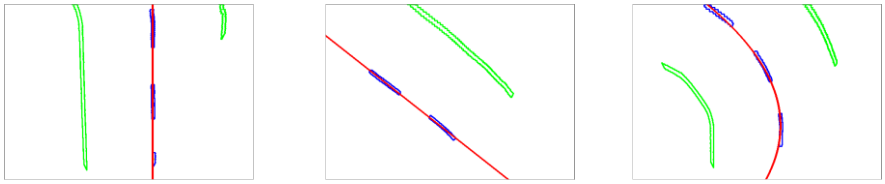
$$H_{sim} = \begin{bmatrix} 640 & -319.561 & -24105.252 \\ 0 & -206.176 & 140282.259 \\ 0 & -0.999 & 564.671 \end{bmatrix}. \tag{2}$$

En la figura 3 se puede ver el resultado de la aplicación de la corrección de perspectiva en la plataforma real con una imagen de la pista. El ancho de cada carril es de 40cm y la línea discontinua central tiene una medida de 20cm en cada segmento al igual que los espacios, lo que puede observarse con ayuda de la cuadrícula verde.



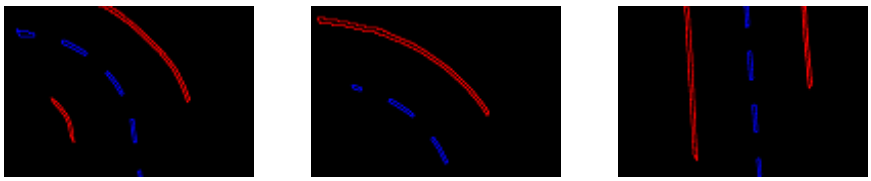
(a) Imagen original (b) Imagen corregida

Fig. 3. Aplicación de corrección de perspectiva.



(a) Polinomio grado 0 (b) Polinomio grado 1 (c) Polinomio grado 2

Fig. 4. Aproximación polinomial.



(a) Detección curva (3 líneas) (b) Detección curva (2 líneas) (c) Detección recta (3 líneas)

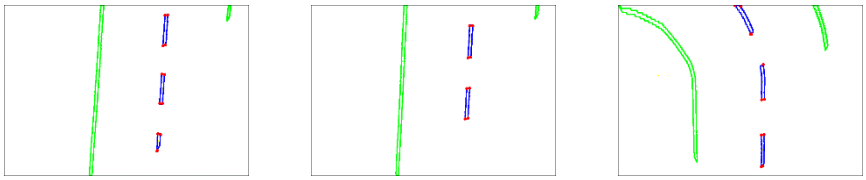
Fig. 5. Detección de líneas de carril en 3 posiciones diferentes.

3.2. Detección de líneas de carriles

La detección de líneas de carriles está enfocada en identificar la línea central y con base en esta clasificar las líneas laterales. La imagen corregida en perspectiva pasa por una etapa de eliminación de píxeles que no son de interés por medio de una operación de binarización con un umbral de 75. Posteriormente se aplica el detector de contornos propio de OpenCV [23] a fin de identificar posibles elementos de las líneas de carril. De estos contornos se eliminan los que tengan áreas muy grandes que corresponden a la porción del suelo visible.

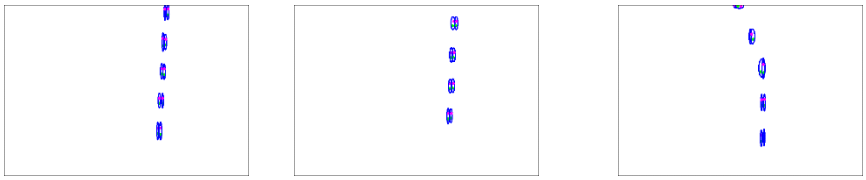
Dentro de esta etapa se realiza una aproximación polinomial, con los centroides de los contornos de la línea central. El polinomio obtenido de esta operación es usado para evaluar los centroides de los contornos nuevos en la imagen siguiente a fin de clasificarlos o no como parte de la línea central. Estos polinomios pueden ser de grado 0, 1 y 2, dependiendo de la ubicación y número de contornos encontrados, como se muestra en la figura 4.

Por ultimo las líneas laterales se identifican según la ubicación respecto a la línea central. La figura 5 muestra los resultados obtenidos en la detección de las líneas de carril, identificando la línea central en color azul y las líneas laterales en color rojo.



(a) Esquinas en 3 componentes (b) Esquinas en 2 componentes (c) Esquinas en curva

Fig. 6. Detección de esquinas.



(a) Asociación en rectas de 3 componentes (b) Asociación en rectas de 2 componentes (c) Asociación en curvas

Fig. 7. Asociación de esquinas.

3.3. Asociación de puntos clave

En esta etapa de la reconstrucción del mapa se busca utilizar las características invariantes en las imágenes consecutivas y que permiten encontrar una relación entre las mismas. En este caso se optó por las esquinas de las componentes de la línea central las cuales, dado el movimiento, no tienen un cambio brusco entre cada imagen sucesiva.

La primera parte de esta etapa corresponde a encontrar las esquinas de las componentes de la línea central. El primer paso es encontrar los coordenadas (x,y) de la imagen que encierran todos los puntos de un contorno en el rectángulo más pequeño posible. Para esto se realiza la búsqueda de las coordenadas (x, y) mayores y menores de cada componente.

Posteriormente se buscan los puntos más cercanos a la combinación de las coordenadas x y y mayores y menores para definir las esquinas de la siguiente forma:

- Esquina 0: es el punto más cercano a x_{menor} y y_{menor}.
- Esquina 1: es el punto más cercano a x_{mayor} y y_{menor}.
- Esquina 2: es el punto más cercano a x_{menor} y y_{mayor}.
- Esquina 3: es el punto más cercano a x_{mayor} y y_{mayor}.

La figura 6 muestra el resultado de esta etapa en tres situaciones distintas, con presencia de componentes centrales completos e incompletos.

Una vez detectadas las esquinas de cada contorno central, lo siguiente es relacionarlas con las esquinas de la imagen anterior. Para este paso, se tiene como consideración el hecho de que entre dos imágenes consecutivas el desplazamiento y rotación no aumentan, o disminuyen según el caso, de manera considerable.

La primera comprobación que se realiza es que el componente central este completo. Dado que estos pueden ser rectos o curvos, se optó por usar la distancia entre la Esquina 0 y la Esquina 3. Si esta distancia es aproximadamente igual a la distancia de un componente completo se utilizan las cuatro esquinas. Si este no es el caso solo se guardan las dos superiores o las dos inferiores, según la ubicación de la componente. Se verifica la distancia entre cada una de las componentes del frame anterior para relacionarlo con el nuevo. En la figura 7 se pueden observar las esquinas relacionadas entre sí. Los óvalos encierran a las esquinas presentes en dos imágenes sucesivas.

3.4. Reconstrucción del mapa

Encontradas las esquinas presentes en dos imágenes consecutivas, así como su relación, el paso siguiente es encontrar la matriz de transformación entre ellas. Para esta etapa se utilizó una variación del método descrito en [24], donde realizan el ajuste de dos colecciones de puntos tridimensionales por medio de una transformación de similaridad. La variación del método consiste en su implementación con dos colecciones de puntos 2D, como se muestra en el Algoritmo 1.

Después de varias pruebas se decidió utilizar la medición de la orientación proporcionada por el sensor IMU. De esta forma cada vez que se captura una imagen nueva se estima la nueva rotación a partir de la resta de la orientación nueva menos la anterior. Esta información se publica en forma de mensajes en tiempo real en el tópico de ROS */model_car/yaw* propio de la plataforma.

Obtener el incremento en la orientación por cada frame capturado requiere de ciertas consideraciones. Esto es debido a la forma en que el sensor IMU otorga la medición del ángulo. Los ángulos medidos están dados en grados y van de 0º a -180º y de 180º a 0º. Debido a esto se realiza una conversión entre el valor otorgado por el sensor y la medición real, además de que la posición del IMU al iniciar el recorrido no es cero.

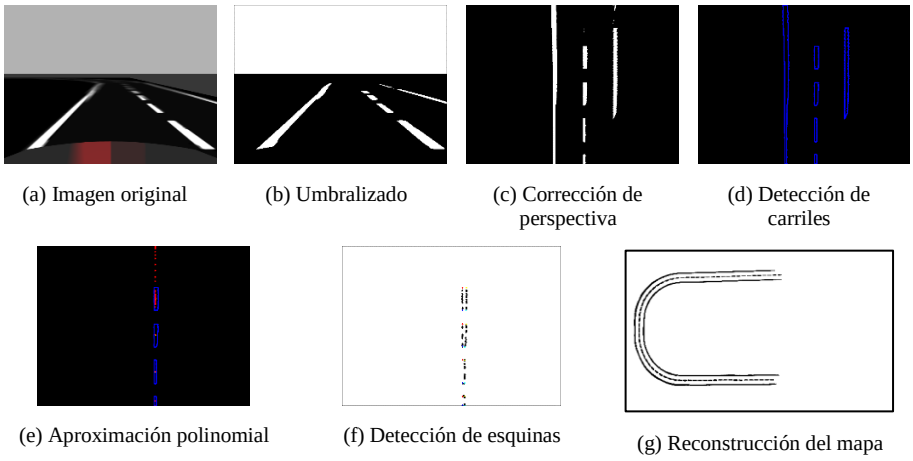
Algoritmo 1. Calculo de matriz de transformación entre dos conjuntos de puntos 2D.

Entrada: Matriz de puntos $A^{2\times n}$ y $B^{2\times m}$
Salida : H: Matriz de transformación

1 $\boldsymbol{P_{m1}} =$ punto medio de A.
2 $\boldsymbol{P_{m2}} =$ punto medio de B.
3 $\boldsymbol{A_m} = \boldsymbol{A} - \boldsymbol{P_{m1}}$
4 $\boldsymbol{B_m} = \boldsymbol{B} - \boldsymbol{P_{m2}}$
5 D = $\boldsymbol{A_m} \cdot \boldsymbol{B_m^T}$
6 $(\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{v_t}) = \boldsymbol{SVD}(\boldsymbol{D})$
7 $\boldsymbol{R} = \boldsymbol{v_t^T} \cdot \boldsymbol{U^T}$
8 $\boldsymbol{t} = -\boldsymbol{R} \cdot \boldsymbol{P_{m1}} + \boldsymbol{P_{m2}}$
9 $\boldsymbol{H} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^{1\times 2} & \boldsymbol{1} \end{bmatrix}$

4. Resultados

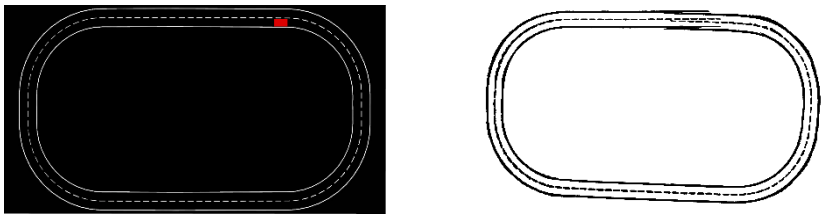
La reconstrucción de mapas se llevó a cabo tanto en el ambiente virtual en Gazebo con tres pistas diferentes y en una sola pista real. En esta última se usaron las imágenes



(a) Imagen original (b) Umbralizado (c) Corrección de perspectiva (d) Detección de carriles

(e) Aproximación polinomial (f) Detección de esquinas (g) Reconstrucción del mapa

Fig. 8. Proceso de construcción del mapa, frame 2000.



(a) Pista virtual 1 (b) Mapa 1

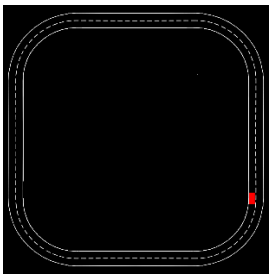
Fig. 9. Reconstrucción del mapa del ambiente virtual 1.

capturadas por la cámara Intel Real Sense RGB de tamaño 640×480 a 60 fps, a bordo de la plataforma AutoNOMOS mini. El vehículo fue teleoperado con un nodo en ROS con una velocidad máxima de 300 rpm, equivalente a 0.177 m/s. Las mediciones de la orientación fueron obtenidas del sensor IMU 6050 a bordo del robot.

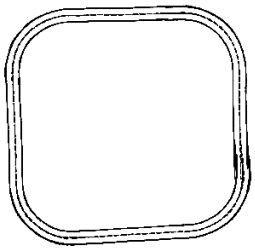
En la figura 8 se observan las etapas del procesamiento, así como la porción reconstruida del mapa al recorrer aproximadamente la mitad de la pista virtual con 2000 imágenes procesadas en tiempo real.

En los escenarios virtuales la reconstrucción del mapa tiene menos errores de segmentación con respecto a la pista real. El error final del ángulo de salida en orientación en la pista real fue menor a 7º lo que es notorio en el emparejamiento del inicio y fin del recorrido.

La reconstrucción completa de dos pistas virtuales se muestra en la figura 9 y 10, donde el recuadro rojo marca el inicio y fin de cada recorrido. De igual forma en la figura 11 se observa la reconstrucción de la pista real. En esta última se observan discontinuidades en las líneas laterales que corresponden a pérdida de información por el campo de visión más pequeño de la cámara de la plataforma real.



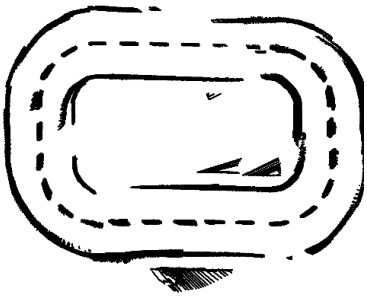
(a) Pista virtual 2.



(b) Mapa 2.

Fig. 10. Reconstrucción del mapa del ambiente virtual 2.

(a) Pista real.



(b) Mapa real.

Fig. 11. Reconstrucción del mapa de la pista real.

Al detectar al menos dos componentes de la línea central la asociación de puntos es invariante a cambios de carril. Los cambios bruscos de dirección al conducir la plataforma influyen en los errores en la medición de la orientación, lo cual afecta a las transformaciones entre imágenes.

En el procesamiento de imágenes de la pista real se tienen errores de segmentación debido al fondo en el que está montada la pista, cuyo color es muy cercano al blanco. Estos errores agregan pixeles erróneos a la reconstrucción, sin embargo visualmente son pocos en comparación a los correctos. Como se puede observar en la 11(b) en algunas partes de las líneas derecha e izquierda hay discontinuidades. Estas se deben a errores de clasificación en líneas rectas y a perdida de información por el campo de visión de la cámara.

5. Conclusiones

Utilizando imágenes capturadas en conjunción con la medición de orientación a bordo de la plataforma AutoNOMOS mini se obtiene una representación gráfica de una pista de competencias. Esto es posible usando técnicas de procesamiento de imágenes y asociación de datos. La corrección de perspectiva tiene un correcto funcionamiento en los dos ambientes, real y virtual. Se implementó un detector de esquinas basado en la forma de las componentes centrales, el cual permite relacionar frames consecutivos.

La asociación de datos se basa en la medida de distancia recorrida calculada en pixeles a partir de las imágenes. Los mapas obtenidos son considerablemente mejores en el ambiente virtual, donde no hay cambios de iluminación y las tonalidades del suelo no difieren de la pista.

Agradecimientos. E. Diaz-Sarmientos agradece a CONACYT por la beca otorgada para la realización de sus estudios de maestría. Agradece también al Centro de Investigación en Computación del IPN por la oportunidad para realizar una estancia de investigación en el Laboratorio de Robótica y Mecatrónica. H. Sossa agradece al Instituto Politécnico Nacional por el apoyo financiero para la realización de esta investigación en el marco del proyecto 20200630.

Referencias

1. Badue, C., Guidolini, R., Carneiro, R.V., Azevedo, P., Cardoso, V.B., Forechi, A., Jesus, L., Berriel, R., Paixão, T.M., Mutz, F., et al.: Self-driving cars: A survey. Expert Systems with Applications (2020)
2. Aly, M.: Real time detection of lane markers in urban streets. In: Intelligent Vehicles Symposium, pp. 7−12 (2008)
3. Lopez, A., Serrat, J., Canero, C., Lumbreras, F., Graf, T.: Robust lane markings detection and road geometry computation. International Journal of Automotive Technology, 11, pp. 395−407 (2010)
4. Guo, C., Mita, S., McAllester, D.: Lane detection and tracking in challenging environments based on a weighted graph and integrated cues. In: Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on, pp. 5543−5550 (2010)
5. Bae, J.H., Song, J.B.: Monocular vision-based lane detection using segmented regions from edge information. In: Ubiquitous Robots and Ambient Intelligence (URAI), 8th International Conference on, pp. 499−502 (2011)
6. Kücükyildiz, G., Ocak, H.: Development and optimization of a DSP-based real-time lane detection algorithm on a mobile platform. Turkish Journal of Electrical Engineering & Computer Sciences, 22, pp. 1484−1500 (2014)
7. Jo, K., Sunwoo, M.: Generation of a precise roadway map for autonomous cars. IEEE Transactions on Intelligent Transportation Systems, 15, pp. 925−937 (2014)
8. Guo, C., Kidono, K., Meguro, J., Kojima, Y., Ogawa, M., Naito, T.: A low-cost solution for automatic lane-level map generation using conventional in-car sensors. IEEE Transactions on Intelligent Transportation Systems, 17, pp. 2355−2366 (2016)
9. Gwon, G.P., Hur, W.S., Kim, S.W., Seo, S.W.: Generation of a precise and efficient lane-level road map for intelligent vehicle systems. IEEE Transactions on Vehicular Technology 66, pp. 4517−4533 (2017)
10. Carolo cup: https://wiki.ifr.ing.tu-bs.de/carolocup/en/carolo-cup (2020)
11. Talent Land Competencias: https://robotics.talent-network.org/edicion-2019/ (2019)
12. AutomodelCar Torneo Mexicano de Robótica 2019: https://www.femexrobotica.org/tmr2019/portfolio-item/autos-autonomos/ (2019)
13. Karouach, I., Ivanov, S.: Lane detection and following approach in self-driving miniature vehicles (2016)
14. Rehder, E., Albrecht, A.: Submap-based slam for road markings. In: 2015 IEEE Intelligent Vehicles Symposium (IV), IEEE, 1393–1398 (2015)

15. Bravo Conejo, C.G.: Navegación autónoma de un robot tipo automóvil en pista de carreras con obstáculos. Master's thesis, CIC IPN (2018)
16. Vanegas Sánchez, T.D.: Navegación autónoma de un vehículo a escala por medio de redes neuronales profundas. Master's thesis, CIC IPN (2018)
17. AutoNOMOS: https://github.com/automodelcar/automodelcarwiki/wiki/hardware-(autonomos-model-v2) (2017)
18. ROS: http://www.ros.org/about-ros/ (2018)
19. ITAM: Ek autonomos sim. https://github.com/eagleknights/autonomos (2019)
20. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003)
21. de la Escalera, A., Armingol, J.M., Pech, J.L., Gómez, J.J.: Detección automática de un patrón para la calibración de cámaras. Revista Iberoamericana de Automática e Informática Industrial RIAI 7, 83–94 (2010)
22. Isern Gonzalez, J.: Estudio experimental de métodos de calibración y autocalibración de cámaras (2003)
23. Suzuki, S. et al.: Topological structural analysis of digitized binary images by border following. Computer Vision, Graphics, and Image Processing, 30(1):32–46 (1985)
24. Arun, K.S., Huang, T.S., Blostein, S.D.: Least-squares fitting of two 3-D point sets. IEEE Transactions on pattern analysis and machine intelligence, pp. 698–700 (1987)