

Detecció del Espacio Libre Conducible

Jose Escribano López

11/06/2016

Resumen– La detección del espacio libre conducible es utilizada en la actualidad tanto para la asistencia a la conducción como para el desarrollo de sistemas de conducción completamente autónomos. Habitualmente, este problema se afronta determinando la profundidad en la imagen mediante sensores (LIDAR) o cámaras estéreo. Este trabajo desarrolla una solución para la estimación del espacio libre conducible mediante el análisis de imágenes generadas con una cámara monocular. Inspirándose en una solución propuesta anteriormente, basada en el uso de técnicas de programación dinámica y la valoración de características en una imagen, este trabajo propone una solución escalable a este problema. Para ello se analiza el uso de características geométricas basadas en contornos y apariencia. Por último se muestran resultados de dicha solución para muestras de imágenes del conjunto KITTI para retos orientados a la conducción autónoma.

Palabras Clave– Espacio libre conducible, Conducción autónoma, Cámara monocular

Abstract– Drivable detection space is currently used for driving assistance and for the development of fully autonomous driving systems. Typically, this problem is tackled by determining the depth in the image through sensors (LIDAR) or stereo cameras. This paper develops a solution for the estimation of free space drivable by analyzing images generated with a monocular camera. Inspired by a solution previously proposed, based in dynamic programming techniques and assessment of features in an image, this paper proposes a scalable solution to this problem. Algorithm use geometric characteristics like appearance and contours based analyzes. Finally we test the results with KITTI road dataset for autonomous driving.

Keywords– Free Space Detection, Autonomous driving, Monocular camera



1 INTRODUCCIÓN

EN la actualidad, unos de los mayores retos de ingeniería es conseguir, mediante la utilización de técnicas de diferentes campos, un vehículo auto-conducido. Un vehículo capaz de imitar las capacidades humanas mediante el análisis del entorno y la toma de decisiones de tal forma que no necesite de conductor para circular. Son numerosas las empresas de reconocimiento mundial implicadas en la investigación de técnicas para este fin como por ejemplo: BMW, Renault, Ford o Google. Por este motivo encontramos una gran evolución en las formas en las que se afrontan las diferentes problemáticas que incluyen la conducción autónoma.

En este trabajo se afronta la detección del espacio libre conducible, la cual se define como: ser capaz de definir el área en la que un objeto en movimiento (nuestro vehículo) puede moverse sin chocar con otro, ver Fig. 1. Habitualmente para dicho problema se recurre a determinar la profundidad en la imagen mediante sensores (como por ejemplo LIDAR) o cámaras estéreo, en los trabajos [4, 5, 6, 7] se muestran diferentes propuestas de solución a dicho problema. Sin embargo trataremos de dar solución mediante el análisis de imágenes generadas con una cámara monocular con el objetivo de reducir el coste y la complejidad del sistema final. Para ello este trabajo se inspira en una solución propuesta anteriormente, basada en el uso de técnicas de programación dinámica y la valoración de características en una imagen [2, 3], y propone una solución escalable a este problema analizando el uso de características geométricas basadas en contornos y apariencia. Utilizando como entrada la salida de un clasificador de píxeles imperfecto basado en apariencia, desarrollado en el CVC y comparando las mejoras que esta solución puede aportar para el equipo de investigación ADAS y el proyecto ACDC [1].

-
- E-mail de contacto: joseegif@gmail.com
 - Mención realizada: Computación
 - Trabajo tutorizado por: Antonio Manuel López Peña (Departamento de Ciencias de la Computación)
 - Curs 2015/16



Fig. 1: Ejemplo de solución al problema del espacio libre conducible.

Con el objetivo de mostrar tanto el desarrollo del trabajo como los resultados de este, se muestra: Los objetivos marcados y subobjetivos derivados de este, la metodología y el entorno que se ha seguido para su correcto desarrollo, la explicación de la base de la solución propuesta, el estudio de las características utilizadas, los resultados extraídos mediante el uso del conjunto de imágenes para la detección de carretera KITTI [8] y las conclusiones del trabajo.

2 OBJETIVOS

Con el fin de poder llevar a cabo un buen trabajo de investigación y desarrollo en el ámbito en el que se ha definido este proyecto, el objetivo principal que se ha marcado para este es: El desarrollo y estudio de un algoritmo de detección del espacio de carretera conducible (libre de obstáculos) que utilice imágenes captadas con un sistema de visión monocular, sin hacer explícitamente ni detección de objetos, ni detección de líneas de carril. Además, dicho algoritmo tomará como entrada adicional la salida de un clasificador de píxeles basado en apariencia, desarrollado en el CVC. En la Fig. 2 se muestran ejemplos de las entradas y las salidas esperadas para este. Por último se efectuarán pruebas utilizando una selección de imágenes del conjunto de reto para la conducción autónoma KITTI [8], añadiendo modificaciones que mejoren el funcionamiento de dicha solución y proponiendo mejoras futuras.

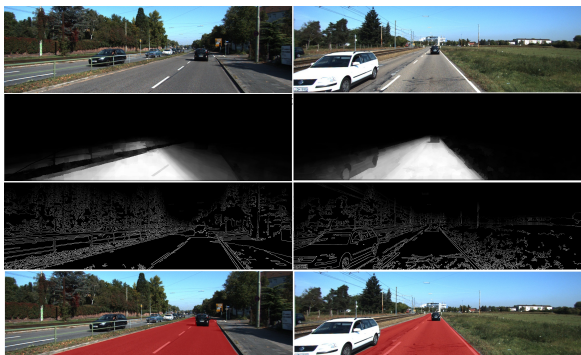


Fig. 2: Primera fila, imagen de entrada al algoritmo. Segunda fila, entrada del clasificador de píxeles imperfecto basado en apariencia. Tercera fila, representación de la función energética utilizada basada en contornos. Cuarta fila, salida ideal esperada para el algoritmo.

Como base del trabajo tomaremos la solución propuesta en el trabajo [2], añadiendo adaptaciones para ajustarlo a las necesidades del trabajo.

2.1. Objetivos derivados

Con el fin de lograr los objetivos principales y para poder hacer un seguimiento más detallado del proyecto, se han definido un conjunto de objetivos derivados de esos, dichos subobjetivos servirán de pauta para el desarrollo del proyecto:

- Estudio inicial del estado del arte para así poder escoger qué soluciones pueden servir de referencia para el proyecto.
- Adquirir los conocimientos necesarios de visión por computador y aprendizaje automático para conducción autónoma. También los entornos de desarrollo con los que se tendrán que trabajar (por ejemplo, la librería OpenCV)
- Implementación de la base funcional del proyecto juntamente con la configuración del entorno seleccionado
- Estudio y integración de las características definidas para el proyecto. Cabe destacar que para el proyecto se ha definido el uso de características geométricas basadas en contornos y apariencia, juntamente con el suavizado espacial definido en el trabajo [2]. Utilizando para la apariencia el clasificador de píxeles desarrollado en el CVC.
- Llevar a cabo mejoras en el algoritmo y en uso de las características que permitan mejorar los resultados que se obtenían anteriormente para la selección del conjunto de imágenes KITTI.
- Consolidar los resultados de la solución proponiendo posibles mejoras de futuro.

3 ESTADO DEL ARTE

Para poder estar informados de las técnicas que actualmente se están llevando a cabo en la dirección de este trabajo, inicialmente se realizó un estudio del estado del arte. Gracias a dicho estudio se observó que las aproximaciones recientes al problema de la detección del espacio libre conducible están enfocadas al uso de cámaras estereo o LIDAR. Algunas de las técnicas utilizadas hasta la fecha y que han demostrado buenos resultados son:

- Detección mediante “Deep learning” con el entrenamiento de redes para dicho fin, como podemos encontrar en el trabajo [5]. En este caso se utiliza una red basada en la arquitectura de la GoogLeNet y OverFeat para la detección de obstáculos y en consecuencia, se extrae el espacio libre conducible.
- Representación del espacio mediante “Stixels”, como se define en el trabajo [7]. Técnica que se basa en la profundidad de la imagen para definir los obstáculos. Utilizando la profundidad de la imagen, se marcan aquellos obstáculos que cuentan con cierto nivel de verticalidad consiguiendo así una representación del espacio 3D del cual se extrae el espacio libre.

A diferencia de estas soluciones, este trabajo afronta la detección del espacio libre conducible mediante el análisis de

imágenes extraídas de una cámara monocular y sin hacer explícitamente detección de objetos. Para ello nos basamos en la solución propuesta en [2] que, mediante la extracción de características geométricas tales como los contornos o la apariencia de la imagen, calcula para cada columna el píxel candidato a ser separador entre espacio libre y espacio no libre. La forma de extracción, clasificación y cálculo del píxel candidato se trata más en profundidad en otras secciones de este documento.

4 METODOLOGÍA Y ENTORNO

4.1. Planificación

Para poder lograr los objetivos mencionados anteriormente, es importante definir una metodología de trabajo adecuada. Dado que el fin de este trabajo es conseguir una solución de utilidad para el proyecto del coche de conducción autónoma del CVC del equipo ADAS [1]. Se ha llevado a cabo una metodología tipo Scrum, realizando reuniones semanales con el tutor para poder hacer el correcto seguimiento del proyecto juntamente con los compañeros del equipo. Esto permite llevar a todo el equipo el seguimiento completo de los módulos que integrarán el coche autónomo mostrando avances en cada uno de los desarrollos. Respecto a la planificación de tareas para el proyecto, este se ha estructurado en un total de cinco fases:

- Fase Inicial, conforma el primer contacto con el proyecto y el problema a analizar. En esta fase se definen los objetivos del trabajo, el entorno en el que se trabajará y se realiza el estudio del estado del arte para conseguir los conocimientos necesarios sobre la materia. Esta fase también agrupa la configuración inicial del entorno elegido. La duración estimada para esta es de una semana y media de trabajo.
- Fase 1, en esta fase se diseña e implementa la primera base funcional de la solución escogida. Se implementa una primera versión de la solución a proponer basándose en el algoritmo propuesto en [2]. Finalmente se analizan los primeros resultados y se aplican las mejoras a la base que se crean convenientes. Esta fase tiene una duración aproximada de un mes.
- Fase 2, en esta fase se integra la salida del clasificador de píxeles imperfecto basado en apariencia con el fin de mejorar los resultados obtenidos hasta el momento. Para ello se estimó un total de unos 25 días de trabajo para poder hacer los estudios y las consideraciones convenientes en la integración.
- Fase 3, se define la preparación del sistema diseñado hasta la fecha para poder hacer el aprendizaje del modelo SVM. El estudio de las características y las relaciones entre ellas con el fin de definir un correcto uso de las funciones de energía en el algoritmo. Y finalmente el desarrollo del aprendizaje de pesos para dichas funciones siguiendo la estructura que se haya estudiado más conveniente. El tiempo estimado para ello es de unos 20 días aproximadamente.
- Fase Final, se cierra el proyecto, se hacen las últimas mejoras sobre el sistema para mejorar algunos aspectos

(como la ventana modificable) y se extraen los resultados de la solución final. Finalmente se procede a la escritura de la documentación en referente al trabajo. Para estas tareas el tiempo aproximado de trabajo es de un mes.

En la Fig. 3 se muestra el diagrama de Gant de las fases principales definidas para el proyecto, sin entrar en las sub-tareas que se definieron para cada fase.

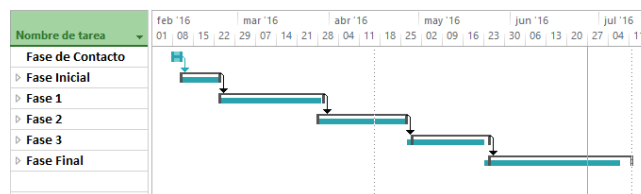


Fig. 3: Diagrama de gant del proyecto en el cual se muestran las fases principales definidas sin mostrar las subtareas de las fases.

4.2. Entorno

El entorno en el que se ha desarrollado el proyecto ha sido decidido por el hecho de que, el sistema final del coche autónomo deberá estar desarrollado para Robotic Operative System (ROS). Teniendo esto en cuenta se ha trabajado con un entorno Ubuntu 14.04, la implementación se ha hecho con Qt5 en C++ y se ha utilizado la librería OpenCV 2.4. Todas las versiones fueron escogidas por el equipo para de esta forma trabajar todos en el mismo entorno y poder así en un futuro hacer la unión de los diferentes sistemas de forma más fácil.

5 BASE DEL ALGORITMO

Este trabajo presenta una solución para la estimación del espacio libre conducible mediante el análisis de imágenes generadas por una cámara monocular. Para ello se ha utilizado la idea presentada anteriormente en el trabajo [2] adaptándola para el objetivo aquí presentado. Así pues se expresa el problema como: conseguir inferir una cadena de Markov, donde cada nodo representa una columna en nuestra imagen y su valor la coordenada vertical del píxel que separa el espacio libre de los obstáculos. Esta cadena por lo tanto nos creará una curva 1D en la imagen donde todo espacio por debajo de ella será el espacio que consideraremos espacio libre conducible, ver Fig. 4.

Para la estimación de la coordenada vertical buscada en cada columna, se define un conjunto de funciones energéticas basadas en características geométricas de la imagen de entrada. Mediante una combinación de dichas funciones, que aprenderemos mediante un modelo utilizando una máquina de vectores de soporte (SVM), se estima en orden (columna por columna) que píxeles son más válidos para ser escogidos como separación.

Este trabajo utiliza la característica de los contornos de la imagen, la apariencia y una tercera función que, basándose en la elección de la columna anterior mantiene el suavizado espacial de la curva que queremos estimar, es por esta última por la que necesitamos hacer el cálculo columna por columna y en orden. A continuación se exponen más

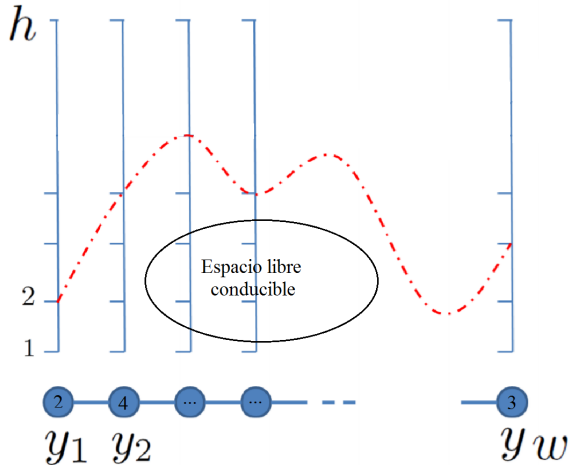


Fig. 4: Representación de la cadena de Markov generada para nuestro algoritmo en la cual Y_i representa el nodo i -ésimo de la cadena, h la altura de la imagen de entrada al algoritmo y w la anchura. Imagen modificada del trabajo [2].

en profundidad las funciones utilizadas, así como el estudio necesario de estas para poder definir un uso correcto.

6 ANALISIS DE LAS CARACTERÍSTICAS

Puesto que nuestro algoritmo se basa en la obtención y valoración de diferentes características de la imagen para poder estimar la línea de separación entre espacio, es importante ver cómo hacemos la definición de estas para nuestra solución. A continuación se definen las funciones de potencial que se han utilizado para nuestra solución así como el estudio que permite extraer la información necesaria para el correcto uso y definición de dichas funciones.

6.1. Contornos

Esta función potencial valora el hecho de que el píxel que genera la separación coincida con un contorno de la imagen. Ya que en una imagen se detectarían numerosos contornos los cuales no serían una buena solución, se valoran más aquellos contornos más cercanos a la cámara. Esto se basa en que es muy posible que los contornos sean generados por obstáculos y por lo tanto, si analizamos la imagen desde la parte inferior, es más posible que los primeros contornos que encontremos sean los que coincidan con la separación entre espacio libre y obstáculos. Para esta valoración se define una función potencial acumulativa tal que:

$$\phi_{edge}(y_i = k) = e(i, k) \sum_{j=k}^h e(i, j) \quad (1)$$

Definiendo $e(i, j)$ como:

$$e(i, j) = \begin{cases} 1 & \text{si el píxel } (i, j) \text{ es contorno} \\ 0 & \text{otro caso} \end{cases} \quad (2)$$

Finalmente se normaliza la función potencial para permitir una mejor clasificación con la SVM y por lo tanto poder aprender un buen modelo. Cabe destacar que este trabajo utiliza el detector de contornos Canny con los parámetros

por defecto pese a que sería posible utilizar un detector más sofisticado con lo que mejoraría la robustez general del sistema.

6.2. Apariencia

Para esta función se utiliza la entrada de un clasificador de píxeles imperfecto basado en apariencia. Este clasificador, mediante la técnica de “superpíxeles” genera una salida en la cual se le asigna a cada píxel la probabilidad de ser o no carretera. La solución que presenta este trabajo utiliza dicha salida para crear una función potencial que nos ayudará a la estimación del píxel que crea la separación del espacio que buscamos. Puesto que la ponderación de los pesos de estas funciones se hará aprendiendo un modelo de máquina de vector de soporte (SVM), se hizo un estudio de cómo definir esta función con la entrada del clasificador y se encontró que la mejor definición era la siguiente:

- Mediante un estudio de la entrada del clasificador de píxeles imperfecto, se define un valor para el que los píxeles con menor probabilidad se encuentran fuera del espacio conducible y los de mayor probabilidad dentro.
- Con el valor definido como “frontera”, y la entrada del clasificador imperfecto, se genera una curva 1D similar a la definida para nuestra solución. Esta curva será la que utilizamos para valorar los resultados del clasificador de apariencia.
- Por último definimos nuestra función potencial basándonos en la cercanía de los píxeles a dicha curva, utilizando la función definida a continuación.

$$\phi_{Ap}(y_i) = \begin{cases} \exp(-\alpha(y_i - a_i)^2) & \text{si } |y_i - a_i| \leq T \\ \lambda & \text{otro caso} \end{cases} \quad (3)$$

Donde a_i sería el valor de la posición i -ésima de la cadena generada para la apariencia. T , α y λ son constantes en nuestro sistema. En la Fig.5 se muestran tanto la entrada del clasificador como una representación de la función potencial para un caso específico.

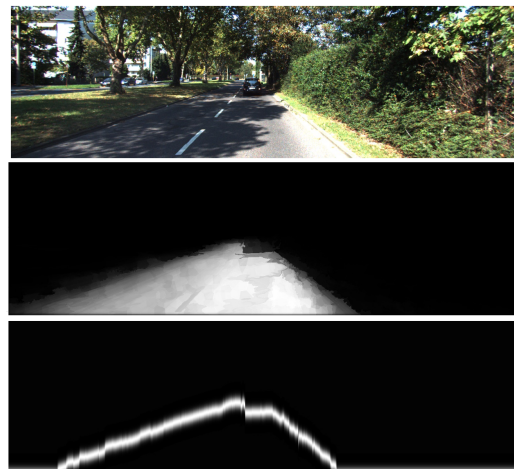


Fig. 5: Primera fila, imagen de entrada original. Segunda fila, salida del clasificador de píxeles imperfecto. Tercera fila, representación de la función potencial definida anteriormente para la entrada de la segunda fila.

6.3. Suavizado espacial

Esta función potencial trata de mantener la continuidad en la curva 1D que buscamos. A partir de la elección que se ha tomado para la columna anterior definimos una función que valora los píxeles más cercanos en la columna actual a dicho valor. El objetivo es lograr mantener la continuidad normal de la frontera de espacio libre, ya que esta solo encontrará saltos bruscos al encontrar un obstaculo que por lo general, son muy pocas columnas en la imagen. La función potencial se define como:

$$\phi S(y_i, y_j) = \begin{cases} \exp(-\alpha(y_i - y_j)^2) & \text{si } |y_i - y_j| \leq T \\ \lambda & \text{otro caso} \end{cases} \quad (4)$$

Donde y_j sería el valor de la cadena generada en la columna anterior a y_i y en $i = 0$ se inicializa $y_j = 0$. Finalmente T , α y λ son constantes en nuestro sistema.

En la Fig.6 se muestran tanto la imagen de entrada como la salida de nuestra solución para dicha imagen y la representación de la función potencial de suavizado espacial que se ha aplicado.

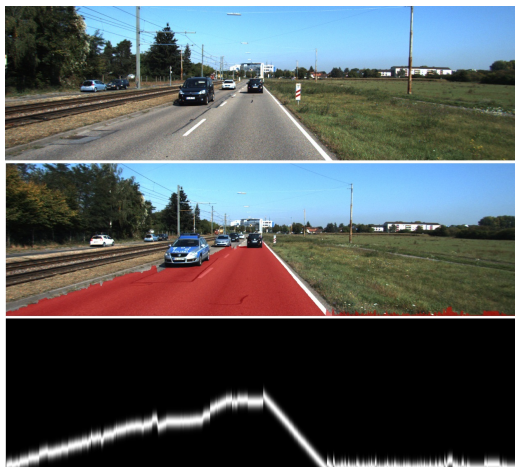


Fig. 6: Primera fila, imagen de entrada original. Segunda fila, salida del algoritmo propuesto. Tercera fila, representación de la función potencial definida anteriormente para el suavizado espacial.

6.4. Relaciones entre las características

Para poder definir cual era la mejor manera de aplicar dichas características en nuestra solución, se hizo un estudio de las relaciones de las características en valores anotados de separación de espacio conducible.

Para dicho estudio se escogieron un conjunto de imágenes anotadas del conjunto para la detección de carretera KITTI [8], en las cuales estaban definidas las líneas que delimitan el espacio conducible, y se mostró qué relación había entre el valor de las diferentes características que hemos definido y los valores de puntos aleatorios de las imágenes que no caían en dicha línea de separación. Con esa información posteriormente se definió cual sería la forma óptima de utilizar estas características para nuestro problema en concreto. Los resultados fueron los siguientes.

Para el estudio de las características de forma individual, se

generaron un conjunto de gráficas comparando el valor que nos retornaba su función potencial para muestras positivas de separación y para muestras negativas. El resultado de dicho estudio fueron los gráficos que se muestran en la fig 7

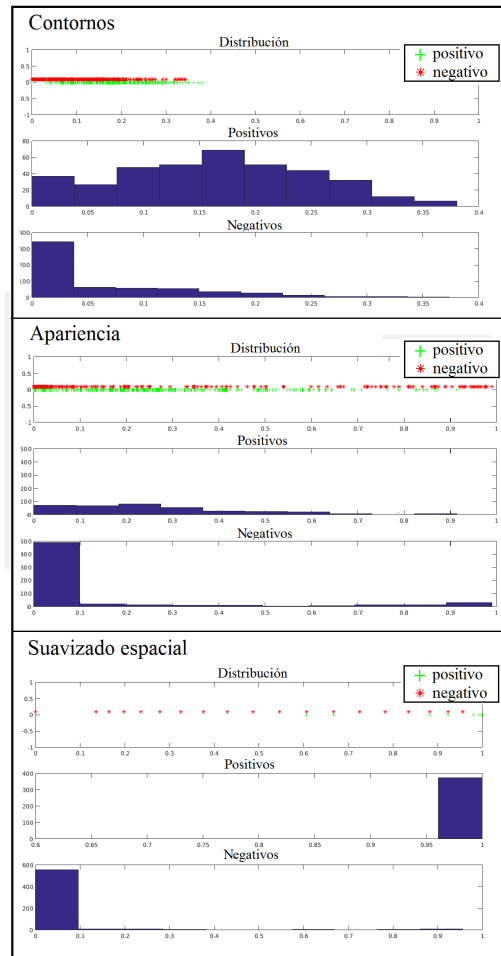


Fig. 7: Gráficos de estudio de las funciones potenciales para casos positivos y negativos generado con 300 imágenes del conjunto KITTI. En cada sección se muestra: la distribución de casos positivos(verde) y negativos(rojo), y los histogramas de valores de casos positivos y negativos respectivamente. Primera sección, función de contornos. Segunda sección, función de apariencia y tercera sección función de suavizado espacial.

Se observa que, a causa de las naturalezas de las funciones definidas para la característica de contornos y apariencia, es posible generar el modelo de SVM para clasificar y por lo tanto ponderar dichas funciones. Sin embargo, ya que la función de suavizado espacial depende de las elecciones tomadas provocaría un modelo erróneo orientándose excesivamente en dicha función potencial. Es por esto por lo que se decidió que el modelo SVM se aprendería sin tener en cuenta el suavizado espacial para posteriormente aplicarlo al resultado.

De esta forma podemos crear un modelo correcto que se base en los contornos y la apariencia, teniendo la posibilidad en el futuro de aplicar más funciones que exploten otras características de la imagen.

Finalmente se generaron representaciones de cómo se distribuían en el espacio dichas características para poder ob-

servar si efectivamente la afirmación expuesta anteriormente era correcta, ver fig. 8.

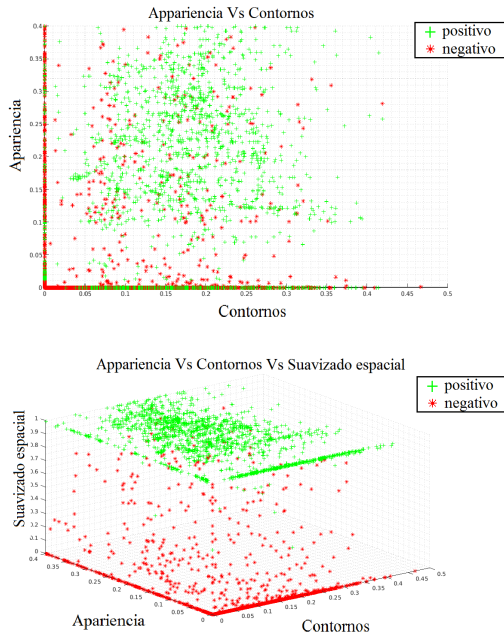


Fig. 8: Gráficos de estudio de las funciones potenciales para casos positivos y negativos generado con 300 imágenes del conjunto KITTI. Se muestran la distribución de valores en el espacio de las funciones tanto para muestras positivas (verde) como muestras negativas (rojo).

7 SOLUCIÓN PROPUESTA

Con el estudio de las relaciones entre las características observado anteriormente, se hicieron un conjunto de pruebas para encontrar cual sería la forma de aplicar las funciones potenciales de forma más correcta. Como se ha observado, la naturaleza de la función utilizada para el suavizado espacial no permite generar un modelo correcto de SVM por lo que se definió el algoritmo de la siguiente forma.

Inicialmente se aprende un modelo SVM, con un conjunto de imágenes anotadas de KITTI para la clasificación de nuestras funciones de contornos y apariencia definidas anteriormente (el algoritmo permite añadir más características en este punto como se comenta en el apartado de propuestas de mejora). Con la SVM entrenada para proporcionar los pesos de las diferentes funciones tendríamos el siguiente flujo en nuestro algoritmo.

- Se toma como entrada la imagen y la salida del clasificador de píxeles imperfecto basado en apariencia.
- Se calcula la función de energía utilizando los pesos del modelo SVM como se muestra en la fórmula 5.
- Añadimos a la función de energía el valor de la función potencial de suavizado temporal para forzar continuidad en el resultado como se muestra en la fórmula 6.
- En cada columna se limita la búsqueda de la solución en referente a la curva 1D que extraemos de la entrada

del clasificador de píxeles imperfecto. Así pues se busca el valor máximo de la función energética E_2 en una altura $a_i \pm V$, siendo a_i el valor de la posición i -ésima de la curva de apariencia y V una constante en nuestro sistema.

- Finalmente, para eliminar posibles errores que puedan provocarse por la limitación del espacio de búsqueda, si la posición resultante difiere mucho con la posición escogida para la columna anterior se aumenta la ventana de búsqueda para eliminar los valores erróneos que provienen del clasificador de píxeles. Esto se hace de manera iterativa multiplicando el valor de V por 2 hasta llegar a eliminar la ventana si es necesario.

Las funciones energéticas utilizadas son:

$$E_1(y, I) = \sum_{n \in N} \sum_i w_u^T \phi_u(y_i) \quad (5)$$

Donde $w = \{w_{edge}, w_{ap}\}$ en nuestro caso y corresponde a los pesos de las funciones de potencial aprendidos mediante el modelo SVM. N es el conjunto de funciones utilizadas para la clasificación y ϕ_u corresponde a la función potencial en cuestión.

$$E_2(y, I) = E_1(y, I) + W_s \phi S(y_i, y_{i-1}) \quad (6)$$

Siendo W_s una constante en nuestro sistema.

8 MÉTRICA DE EVALUACIÓN

Con el fin de poder evaluar los resultados correctamente, se ha definido una fórmula basada en la diferencia de píxeles que se encuentra entre las anotaciones reales de las imágenes y la salida de nuestro algoritmo. Este método es similar al utilizado para la valoración de los resultados que se muestra en el trabajo [2], y consiste en calcular la diferencia relativa existente entre la estimación y el resultado ideal utilizando la fórmula 7.

$$DR(y, y^*) = \frac{|y - y^*|}{n * h} \quad (7)$$

Donde y es la curva estimada de nuestro algoritmo, y^* corresponde a la curva del “ground truth”, h es la altura de la imagen y n es la constante de normalización, la cual valdrá el número de columnas de la imagen. De esta forma $DR(y, y^*)$ nos proporciona un valor entre 0 y 1 que corresponde al porcentaje de la imagen que representa la diferencia entre la respuesta ideal y la estimada.

9 RESULTADOS

En este apartado analizaremos los resultados de nuestra solución. Para ello se ha utilizado una selección del conjunto de imágenes KITTI para retos orientados a la conducción autónoma. Se mostrarán los resultados del algoritmo para diferentes configuraciones y etapas del proyecto. En la fig. 9 se muestran un conjunto de entradas y salidas del algoritmo juntamente a la representación de la función de energía generada en cada caso.

A continuación se muestran los resultados para la métrica

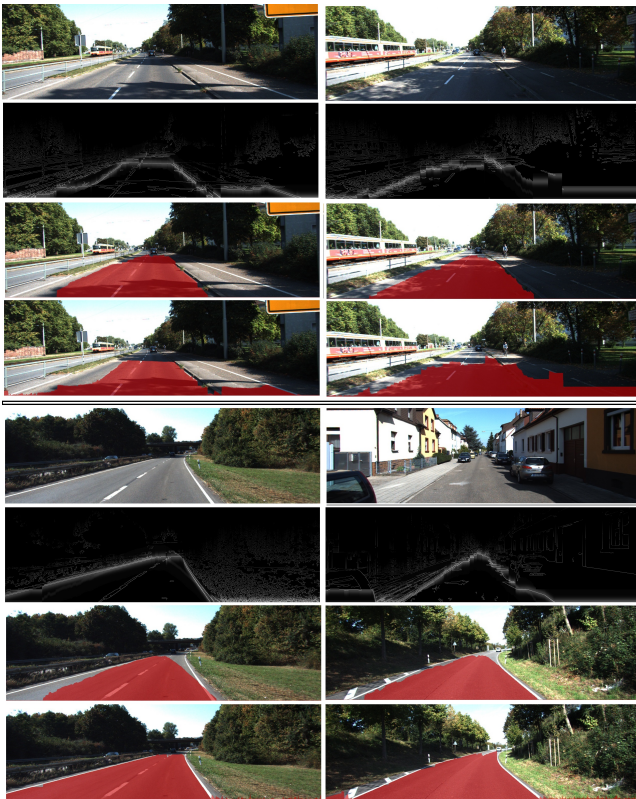


Fig. 9: Conjunto de análisis de 4 imágenes con el algoritmo propuesto. Para cada imagen se muestra: Primera fila, imagen de entrada al algoritmo; Segunda fila, representación de la función de energía utilizada por el algoritmo para la imagen de entrada; Tercera fila, salida de la curva del clasificador de píxeles basado en apariencia; Cuarta fila, salida de la curva estimada por la solución propuesta.

definida anteriormente de un conjunto seleccionado aleatoriamente de 60 imágenes de diferentes características del conjunto KITTI con los diferentes algoritmos:

- Solución: solución final propuesta en este artículo.
- Solución nvm: solución con ventana fija no modificable.
- Solución nv: solución sin ventana de limitación de búsqueda.
- Cadena Ap: solución del clasificador de píxeles.

Algoritmo	Diferencia Relativa
Solución	4.1 %
Solución nvm	4.5 %
Solución nv	5.6 %
Cadena Ap	4.0 %

Al analizar las imágenes se detectó que nuestra solución empeoraba a la del clasificador de píxeles basado en apariencia en aquellos casos en los que existían fuertes sombras transversales o imperfecciones en la carretera. Si miramos la Fig 9, las dos imágenes superiores muestran este caso. La causa de esto es que al basarse en los contornos de la imagen, si dichas sombras o imperfecciones generaban contornos muy pronunciados, el algoritmo los identifica como frontera a espacio libre. Es por esto por lo que se

decidió hacer una subdivisión de las imágenes de muestra para valorar el funcionamiento del algoritmo en casos con sombra pronunciada y en casos con sombras leves o sin imperfecciones en la carretera, ya que estos últimos son casos más comunes en una autopista.

La tabla mostrada a continuación recopila la información de los algoritmos definidos anteriormente para aquellos casos de imágenes en los cuales se encuentran sombras transversales o imperfecciones que perjudican nuestra solución.

Algoritmo	Diferencia Relativa
Solución	6.7 %
Solución nvm	6.3 %
Solución nv	8.1 %
Cadena Ap	5.9 %

Analizando los datos se observa que, pese a que el clasificador de píxeles también se ve perjudicado por estos casos respecto a los resultados generales, nuestro algoritmo se ve mucho más afectado a causa de la función potencial basada en los contornos de la imagen que se utiliza.

La siguiente tabla recopila aquellos casos más similares a los que afrontaría un vehículo por autopista, es decir, imágenes con carreteras sin grandes imperfecciones y sin sombras muy marcadas que puedan afectar al detector de contornos.

Algoritmo	Diferencia Relativa
Solución	3.1 %
Solución nvm	3.3 %
Solución nv	4.5 %
Cadena Ap	4.7 %

Analizando los datos de esta última tabla se observa una clara mejora de la solución propuesta con respecto al clasificador de píxeles basado en apariencia y las diferentes modificaciones planteadas para este algoritmo. Se puede así extraer que la solución muestra buenos resultados con el conjunto de imágenes para retos de conducción autónoma KITTI. Cabe destacar que, pese a que los resultados con las muestras generales demuestran una estimación bastante robusta de la frontera del espacio libre conducible, la solución aquí propuesta permite bastantes mejoras. Dichas mejoras se han definido como posible trabajo futuro para este proyecto.

10 PROPUESTAS DE MEJORA

Pese a que para este trabajo se han invertido numerosas horas en la búsqueda e implementación de mejoras respecto al algoritmo base, como por ejemplo la ventana de búsqueda. La duración limitada de este no ha permitido implementar todas las mejoras que se han pensado posibles.

En esta sección se exponen un conjunto de modificaciones o mejoras que ayudarían a conseguir una estimación más robusta de la solución con el objetivo de continuar mejorando dicha solución en desarrollos posteriores. Las mejoras más significativas planteadas son:

- Cálculo de los mejores valores de constantes: como se ha expuesto en este documento, la solución cuenta de

numerosas constantes como el valor de frontera escogido para la apariencia, las constantes utilizadas para el suavizado espacial, los valores del detector de contornos, entre otros. El cálculo de dichos valores mediante cross-validation mejoraría el funcionamiento y la robustez del sistema.

- Añadir más características geométricas: como se ha expuesto en este trabajo, la forma en la que se trabajan las funciones de potencial permite añadir más características teniendo simplemente que reentrenar el modelo SVM para la ponderación de estas.
- Inserción de suavizado temporal mediante el cálculo de una homografía, comparando la curva generada en el fotograma anterior con la que se quiere estimar. Esto mejoraría la robustez del sistema al tratar secuencias de imágenes.
- Modificación del clasificador SVM utilizando otro o modificando el kernel para permitir una mejor clasificación de las características y por lo tanto una mejor ponderación de las funciones de potencial.
- Integración conjunta con módulos de detección de objetos o líneas de carril para proporcionar una estimación más robusta.
- Control del punto de fuga para la limitación del espacio de búsqueda.

Estas modificaciones permitirían mejorar el sistema sin incluir la profundidad de la imagen por lo que el objetivo del trabajo no se vería modificado.

11 CONCLUSIONES

En este trabajo se ha propuesto una solución escalable al problema de la detección del espacio libre conducible. Dicha solución utiliza imágenes generadas de una cámara monocular y integra la salida de un clasificador de píxeles imperfecto basado en la apariencia.

La solución propuesta utiliza un conjunto de funciones de potencial basadas en la extracción de características de la imagen como los contornos, y mediante un modelo de máquina de vector de soporte (SVM) pondera dichas funciones para conseguir una función de energía que permite estimar el píxel que crea la separación entre el espacio libre conducible de la imagen y los obstáculos. También se han añadido modificaciones para mejorar la robustez del sistema final. Se han demostrado resultados prometedores con una selección de imágenes del conjunto para la detección de carretera KITTI [8]. Mejorando el clasificador de píxeles en casos similares a los de autopista. Y finalmente se han definido un conjunto de mejoras posibles para la solución propuesta a modo de trabajo futuro para el proyecto.

AGRADECIMIENTOS

A todo el equipo ADAS del CVC que me han ayudado en conceptos de visión por computador y funcionalidades utilizadas por el CVC que han sido necesarias para la ejecución de este trabajo, en especial a David Vázquez, Gabriel

Villalonga y Antonio López. Y finalmente a mi compañero José Luis Gómez el cual realizaba otro módulo de detección de líneas de carril, ya que nos ayudábamos mutuamente y me ha acompañado en el desarrollo del trabajo.

REFERENCIAS

- [1] Equipo ADAS de investigación para vehículo de conducción autónoma del CVC y proyecto ACDC. In <http://adas.cvc.uab.es/projects/ACDC/>
- [2] J. Yao, S. Ramalingam, Y. Taguchi, Y. Miki, R. Urtasun. "Estimating Drivable Collision-Free Space from Monocular Video". In <http://www.cs.toronto.edu/yao-jian/freeSpace.pdf>
- [3] J. Yao, S. Ramalingam, Y. Taguchi, Y. Miki, R. Urtasun. [Patent] "Estimating Drivable Collision-Free Space from Monocular Video". In <http://www.patentsencyclopedia.com/app/20150206015>
- [4] H. Badino, U. Franke, and R. Mester. "Free space computation using stochastic occupancy grids and dynamic programming". In *ICCV Workshop on Dynamical Vision*, 2007.
- [5] Joel Pazhayampallil . "Free Space Detection with Deep Nets for Autonomous Driving". Stanford University
- [6] J. Moras, S. A. Rodríguez F., V. Drevelle, G. Dherbomez, V. Cherfaoui, P. Bonnifait. "Drivable Space Characterization using Automotive Lidar and Georeferenced Map Information". IEEE 3-7 June 2012
- [7] H. Badino, U. Franke, and D. Pfeiffer. "The stixel world - a compact medium level representation of the 3d-world." In *DAGM*, 2009.
- [8] J. Fritsch, T. Kuehnl, A. Geiger "A New Performance Measure and Evaluation Benchmark for Road Detection Algorithms". In *International Conference on Intelligent Transportation Systems (ITSC)*, 2013.