

UNIVERSIDAD DE ALCALÁ
Escuela Politécnica Superior

GRADO EN INGENIERÍA EN ELECTRÓNICA Y
AUTOMÁTICA INDUSTRIAL

Trabajo Fin de Grado

Detección de bordillos en curva en escenas urbanas
mediante visión estéreo a partir de la detección de
bordillos rectos

Alumno: D. Álvaro García Morcillo.

Director: D. Miguel Ángel Sotelo Vázquez.

Tribunal:

Presidente: D. David Fernández Llorca.

Tutor/Director: D. Miguel Ángel Sotelo Vázquez.

Vocal 1º: D. Iván García Daza.

Suplente: D. Ignacio Parra Alonso.

Calificación:

Fecha:

A los Canteros,
a mis padres que tanto me aguantan,
y a la persona que más me ha apoyado en mis momentos más difíciles,
el bueno de Nacho.

Agradecimientos

A Miguel Ángel, por la confianza depositada y permitirme realizar este TFG sin nociones previas en visión artificial.

A David, Carlos y Raúl, porque han sido buenos profes y por aguantarme en el laboratorio, se que debe haber sido muy duro.

A Rubén, por guiarme todos estos meses, todos esos cafés y porque sin el todavía estaría intentando hacer mi primera compilación.

y en general a todos los compañeros de *IsisLab*, aunque nos hayamos visto poquito.

A mis padres, por estar insistiendo en que termine esta memoria cada día.

A los Canteros, por seguir unidos y que la familia no nos separe.

A mis amigos de la uni que tanto hemos compartido opiniones, comidas y salidas, aunque algunos roben la patata del día...

Y en especial a mi gran amigo Nacho, sin tu apoyo este hubiera sido un año simplemente terrible.

Gracias a todos.

Índice general

| | | |
|-----------|---|-----------|
| I | Resumen | 1 |
| 0.1. | Resumen | 3 |
| 0.2. | Abstract | 4 |
| II | Memoria | 5 |
| 1. | Introducción | 7 |
| 1.1. | Estructura | 8 |
| 2. | Estado del Arte | 9 |
| 2.1. | Un poco de historia | 9 |
| 2.2. | Un antes y un después, el DARPA Urban Challenge | 10 |
| 2.3. | El material de trabajo, el KITTI DataSet | 12 |
| 2.4. | Algoritmos de detección de bordillos | 12 |
| 2.4.1. | Algoritmo de detección de bordillo con medición láser | 13 |
| 2.4.2. | Algoritmo de detección de bordillo con visión estéreo | 15 |
| 2.5. | Objetivos | 18 |
| 3. | Formación de la perspectiva Vista de Pájaro | 19 |
| 3.1. | Parámetros de una cámara | 19 |
| 3.1.1. | Parámetros intrínsecos | 19 |
| 3.1.2. | Parámetros extrínsecos | 19 |
| 3.2. | Formación de la vista de pajarero | 20 |
| 4. | Mapa de elevación abatido | 23 |
| 4.1. | Análisis previo | 23 |
| 4.2. | Formación del mapa de elevación abatido | 23 |
| 4.2.1. | Extracción del plano principal | 23 |
| 4.2.2. | Obtención del mapa de normales | 25 |
| 4.2.3. | Colocación de elevaciones | 25 |
| 4.3. | Enmascaramiento de regiones | 27 |
| 4.4. | Inclusión de obstáculos | 29 |
| 4.5. | Problema de los postes | 29 |
| 5. | Algoritmo desechado: Ajuste de elipses | 31 |
| 6. | Algoritmo de gusano | 33 |
| 6.1. | Idea general | 33 |
| 6.2. | Criterio de bordillo | 33 |
| 6.3. | Nacimiento | 34 |

| | |
|---|-----------|
| 6.3.1. A partir de bordillo visible | 34 |
| 6.3.2. Ubicado automáticamente | 34 |
| 6.3.3. Ajuste al bordillo | 35 |
| 6.4. Búsqueda | 35 |
| 6.4.1. Geometría | 36 |
| 6.4.2. Criterio de certeza | 37 |
| 7. Resultados | 39 |
| 8. Conclusiones y Trabajos Futuros | 49 |
| 8.1. Conclusiones | 49 |
| 8.2. Trabajos futuros | 49 |
| III Bibliografía | 51 |

Índice de figuras

| | |
|---|----|
| 1.1. Ejemplo de capturas usadas para probar y desarrollar el algoritmo | 7 |
| 2.1. Boss, ganador del DARPA Urban Challenge 2007 | 10 |
| 2.2. Estructura del vehículo autónomo predominante en el DARPA <i>Urban Challenge</i> | 11 |
| 2.3. Vehículo de recogida de datos del KITTI. | 13 |
| 2.4. Resultado del algoritmo con láser | 15 |
| 2.5. A la izquierda el DEM, más oscuro significa más alto, a la derecha el mapa de validez, que enmascara los puntos con información 3D. | 16 |
| 2.6. A la izquierda la máscara circular y a la derecha un ejemplo con datos reales, los píxeles de la línea están dibujados en negro. | 17 |
| 2.7. Un bordillo curvo se aproxima a una serie de segmentos de bordillo encadenados. | 18 |
| 2.8. Resultado de la detección proyectado. | 18 |
| 3.1. Transformación a vista de pájaro | 21 |
| 3.2. Ejemplo de una vista de pájaro utilizada y su máscara | 22 |
| 4.1. Representación xyz en escala de grises, los puntos sin disparidad se saturan | 24 |
| 4.2. Representación de normales | 26 |
| 4.3. Efecto del plano principal | 27 |
| 4.4. Enmascaramiento de regiones | 28 |
| 4.5. Rellenado parcial de información en objetos verticales | 29 |
| 5.1. Parámetros utilizados para el barrido de elipses. | 31 |
| 5.2. Resultado en la vista de pájaro del barrido de elipses. En rojo la que mejor se ajusta según diferencia de medias, en azul según diferencia de medianas. | 32 |
| 6.1. Zonas con componente normal que forma un color anaranjado al representarse en HSV las normales | 35 |
| 6.2. Ángulos de búsqueda del gusano en el lado izquierdo. | 36 |
| 6.3. $\alpha = 0,5$ $\omega = 0,2$ $\sigma_f = 0,6$ $\sigma_n = 0,3$ | 38 |
| 7.1. Imagen de elevaciones y resultado en vista de pájaro, secuencia 1 | 41 |
| 7.2. Imagen de elevaciones y resultado en vista de pájaro, secuencia 2 | 42 |
| 7.3. Imagen de elevaciones y resultado en vista de pájaro, secuencia 3 | 43 |
| 7.4. Imagen de elevaciones y resultado en vista de pájaro, secuencia 4 | 44 |
| 7.5. Imagen de elevaciones y resultado en vista de pájaro, secuencia 5 | 45 |
| 7.6. Imagen de elevaciones y resultado en vista de pájaro, secuencia 6 | 46 |
| 7.7. Imagen de elevaciones y resultado en vista de pájaro, secuencia 7 | 47 |

Parte I

Resumen

0.1. Resumen

Este trabajo tiene como objetivo principal el desarrollo de un algoritmo capaz de realizar un seguimiento del trazado de un bordillo en una escena urbana, concretamente en intersecciones, captada a través de un sistema estéreo de cámaras situado encima de un vehículo cuya información está guardada en una base de datos pública.

Se pretende que sea innovador en cierta medida y estar enfocado en una aplicación específica, utilizando un método distinto a los comúnmente utilizados.

La principal dificultad del trabajo radica en el desconocimiento de cierta información tridimensional como las normales y la curvatura en la parte oculta de los bordillos, además de la falta de información visual claramente identificativa de los mismos.

Palabras clave

- Vista de pájaro.
- Mapa de elevación.
- Plano principal.
- Gusano.

0.2. Abstract

This work has as main goal to develop an algorithm capable of tracking a curb in an urban scene, specifically at an intersection, captured by a stereo camera system on the top of a vehicle which information is stored on a public database.

It is intended to be innovative to some extent and to be focused on an specific application, using a distinct method from the commonly used.

The principal difficulty of the work is located on the unknown of certain tridimensional information such as normal and curvatures in the hidden part of the curb, as well as the not enough visual information clearly identifying the curb.

Keywords

- Bird View.
- Elevation Map.
- Principal plane.
- Worm.

Parte II

Memoria

Capítulo 1

Introducción

En los últimos años el sector del automóvil ha incluido numerosos sistemas de ayuda a la conducción en sus vehículos. Comenzó con el aviso por cambio involuntario de carril, continuó con los sistemas de aparcamiento guiados y hoy en día se dispone desde sistemas capaces de aparcar por si mismos hasta vehículos que se detienen en caso de detectar una posible colisión.

En los tiempos actuales, y gracias tanto al gran avance de las tecnologías en los últimos años como a las numerosas investigaciones en décadas anteriores se están desarrollando multitud de proyectos de investigación dedicados ya no solo a la consecución de sistemas de ayuda a la conducción si no a la obtención de sistemas autónomos de transporte. El trabajo desarrollado tanto por parte de los fabricantes como de empresas privadas o instituciones públicas está dando como resultado sistemas cada vez más complejos, útiles y fiables.

Este trabajo de fin grado se basa en una aplicación específica en detección de bordillos, presentes en escenas urbanas, aplicación de la cual existen varias referencias acerca de trabajos detectándolos pero ninguna ofreciendo una solución para detectar un bordillo del que no se puede ver el propio cuerpo del mismo, siendo en tal caso una primera aproximación a esa tarea específica.

En este documento se desarrolla un algoritmo que permitiría trazar el recorrido de un bordillo visible previamente detectado en su parte no visible si dispone de ella basándose en la geometría de la escena. La determinación de estos bordillos serviría como apoyo a otras aplicaciones, como delimitar más fácilmente el espacio libre de circulación.

Para el desarrollo y pruebas del algoritmo se ha empleado una base de datos pública con información obtenida a través de un sistema de cámaras estéreo debidamente calibrado que se detallará en el siguiente capítulo, se utilizan capturas monocromáticas como las de la figura 1.



Figura 1.1: Ejemplo de capturas usadas para probar y desarrollar el algoritmo

1.1. Estructura

El documento que describe el algoritmo se divide en los siguientes capítulos:

- **Capítulo 1:** Introducción: Se plantea la forma en que se desarrolla el trabajo además de la estructura del documento.
- **Capítulo 2:** Estado del Arte: Se realiza un breve repaso de distintas aportaciones o técnicas llevadas a cabo por distintos grupos de investigación en aplicaciones similares a la planteada en este trabajo, y se enumeran los objetivos que se persiguen con el desarrollo de este TFG.
- **Capítulo 3:** Formación de la perspectiva Vista de Pájaro: Se expone cómo es generada la transformación a vista de pájaro, la cual es clave en el desarrollo del trabajo.
- **Capítulo 4:** Mapa de elevación abatido: Se detalla la formación de un mapa de elevación a través de la vista de pájaro y la reconstrucción estéreo con características significativas.
- **Capítulo 5:** Algoritmo desechado: Se describe la implementación de un algoritmo sencillo que fue desechado a la hora de encontrar el bordillo curvo.
- **Capítulo 6:** Algoritmo de gusano: Se detalla el algoritmo final utilizado para la búsqueda del bordillo a partir del mapa de elevación abatido.
- **Capítulo 7:** Resultados: Se explican los resultados obtenidos tras probar el algoritmo en diferentes secuencias sobre las vistas de pájaro generadas.
- **Capítulo 8:** Conclusiones y Trabajos Futuros: Se resumen las conclusiones obtenidas durante la realización del trabajo, proponiendo ciertas mejoras que pueden plantearse y desarrollarse.

Capítulo 2

Estado del Arte

Para llegar al objetivo del trabajo, se explicará a continuación, de forma breve, desde lo general a lo concreto, lo que antecede al trabajo realizado, dando un vistazo a la historia de la conducción autónoma y el uso de la visión artificial en la misma.

2.1. Un poco de historia

El desarrollo de vehículos inteligentes ha sufrido un rápido progreso en los últimos 30 años. El campo de la conducción autónoma posee una historia muy rica y remonta sus orígenes a los años 20, dando lugar a pruebas prometedoras en los años 50 y a los primeros vehículos autosuficientes verdaderamente autónomos en los años 80, demostrándose así la viabilidad del desarrollo de coches capaces de controlar su propio movimiento en entornos complejos como autopistas y calles.

En 1925, la empresa de radio Houdina Radio Control Co. sacó a relucir en las calles de Manhattan, Nueva York, el primer vehículo sin conductor, que, muy lejos de ser un vehículo autónomo, era controlado remotamente por señales de radio desde un segundo vehículo que le seguía por detrás. Más adelante Archen Motors, un distribuidor de Milwaukee [1] y alrededores hizo otra demostración en las calles de Milwaukee, y en 1932 en las calles de Fredericksburg, Virginia.

En 1958, se realizó una demostración de un desarrollo de RCA Labs en el estado de Nebraska en una autovía pública. Una serie de circuitos detectores enterrados en el pavimento, podían enviar impulsos para guiar el coche y determinar la presencia y velocidad de cualquier vehículo metálico en su superficie. Fue desarrollado en colaboración con *General Motors*.

Más adelante, en 1960 se realizó una demostración en Princeton, Nueva Jersey, donde se permitió a los periodistas montar en los vehículos [2].

En la década de los 80 fue de especial importancia el programa PROMETHEUS (Program for European Traffic with Highest Efficiency and Unprecedented Safety) [3], el cual fue desarrollado entre los años 1986 y 1994, desarrollado por EUREKA (European Research Coordination Agency). Los principales objetivos que se buscaban conseguir eran la mejora del tráfico y la circulación, mayor seguridad en la conducción, reducción del número de accidentes, gestión eficiente de flotas de mercancías y reducción de emisiones contaminantes.

Dentro del proyecto PROMETHEUS se desarrollaron siete campos diferentes:

- Sistemas a bordo para monitorización de vehículos y asistencia al conductor.
- Comunicación entre distintos vehículos.
- Comunicación entre carretera y vehículo para controlar el tráfico.
- Desarrollo de componentes electrónicos requeridos.
- Inteligencia artificial y desarrollo de software
- Comunicación dentro del sistema del vehículo y el conductor.
- Efectos del cambio de vehículos en el entorno del tráfico.

Como resultado de este proyecto surgen nuevos prototipos, sistemas y algoritmos, de entre ellos se destacan VITA [4] y GOLD [5].

Los avances en sensado y tecnologías de computación en ese periodo despertaron el interés en el campo de los vehículos autónomos y muchos avances fueron realizados durante una serie de competiciones a nivel mundial patrocinadas por el Departamento de Defensa de los Estados Unidos de mano del DARPA (*Defense Advanced Research Projects Agency*), comenzaron en el 2004 con el nombre de *Grand Challenge*.

2.2. Un antes y un después, el DARPA Urban Challenge

De las competiciones del *Grand Challenge* la más representativa en el campo en que se basa este trabajo, el urbano, es el DARPA *Urban Challenge* (DUC) realizado en 2007 [6].

De un total de 35 equipos que se presentaron en el evento de cualificación nacional (NQE), sólo 11 lograron continuar hasta el evento del desafío urbano final (UCE). El ganador del DUC fue Boss [7] (Figura 2.1), un Chevrolet Tahoe con sus pertinentes modificaciones que fue desarrollado por el Tartan Racing Team compuesto por estudiantes, miembros e investigadores de varias entidades, incluyendo la Carnegie Mellon University, General Motors, Caterpillar, Continental e Intel.



Figura 2.1: Boss, ganador del DARPA Urban Challenge 2007.

De las arquitecturas de los sistemas de los vehículos que participaron en la prueba se puede abstraer un modelo que descompone el problema en cuatro subsistemas de alto nivel (ver Figura 2.2):

1. Sensado: Responsable de obtener información de medidas.
2. Percepción: Para crear información útil sobre el vehículo y su entorno.
3. Planificado: Algoritmos para computar el camino hacia el objetivo.
4. Control: Actuadores y comandos para manejar el coche.

De estos subsistemas el trabajo se basa en el primero, sensado. En las pruebas, para este subsistema se utilizaban las siguientes herramientas de medida:

- GPS (Sistema de posicionamiento global)
- Lidar (Light Detection and Ranging)
- IMU (Unidad de medida inercial)
- Radar
- Odometría
- Visión

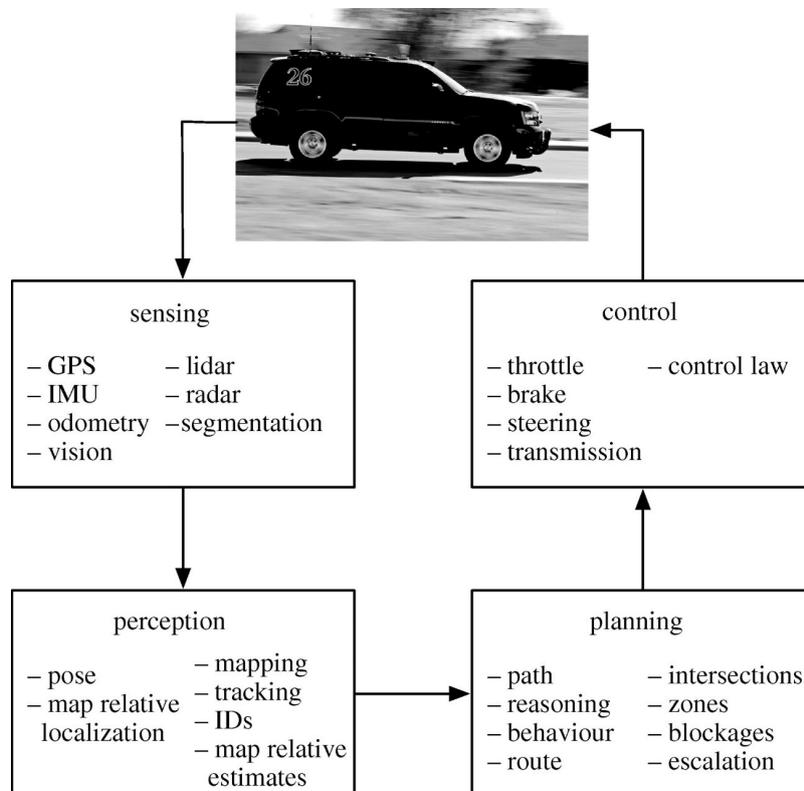


Figura 2.2: Estructura del vehículo autónomo predominante en el DARPA *Urban Challenge*.

De los sistemas de sensado que encontramos nos centramos en la visión artificial, el uso que se le daba a esta tecnología en los campeonatos del DARPA era principalmente la detección de marcas viales para distinguir carriles y líneas de parada, prefiriéndose tecnologías basadas en láser para tratar con información tridimensional. Sin embargo la visión artificial puede complementar al escáner láser, de hecho la consecución del objetivo de este trabajo no podría conseguirse con la información obtenida de un escáner láser, ya que sería insuficiente.

2.3. El material de trabajo, el KITTI DataSet

La visión artificial es un subcampo de la inteligencia artificial utilizado para la detección de carriles, señales, y marcas viales en el caso de la conducción autónoma.

Actualmente la percepción en conducción autónoma se basa principalmente en el uso del GPS, radar y lidar, así como en mapas muy precisos del entorno, pero el instituto de tecnología de Karlsruhe, junto al Instituto tecnológico Toyota de Chicago desarrollaron un proyecto para dar referencias y una base de datos a través de medidas reales, fuera del laboratorio a la comunidad investigadora. Los intereses de este proyecto se basan en el estéreo, el flujo óptico, la odometría visual y la detección de objetos y rastreo tridimensionales. Este proyecto se tituló como *The KITTI Vision Benchmark Suite* [8].

La base de datos formada en este proyecto contiene información recogida por un Volkswagen Passat B6 a través de los siguientes sensores (Ver figura 2.3)

- 2 cámaras monocromáticas PointGrey Flea 2, 1.4 Megapíxeles.
- 2 cámaras a color PointGrey Flea 2, 1.4 Megapíxeles.
- 1 sistema de navegación inercial GPS/IMU OXTS HT 3003.
- 1 escáner laser Velodyne.

De todo lo extraíble de la base de datos el objetivo es realizar el trabajo exclusivamente con la información intrínseca y extrínseca de cada cámara a la par que con los datos tomados por éstas, excluyendo toda la información del sistema de navegación inercial y el escáner laser.

A partir de este punto se trata el tema de detección de bordillos, exponiendo algoritmos tanto con tecnología láser como con visión artificial.

2.4. Algoritmos de detección de bordillos

Los algoritmos que han tenido éxito hasta ahora para la detección de bordillos utilizan información tridimensional. Para obtener dicha información han predominado dos tecnologías: la visión estéreo y la medición láser.

Las principales diferencias entre ambas tecnologías radican en los siguientes factores:

- **Precio:** El coste de un escáner láser es mucho más elevado que el de una cámara estéreo o un par de cámaras normales.
- **Densidad de información:** De la visión estéreo se extraen nubes de puntos más densas que con el escáner láser.

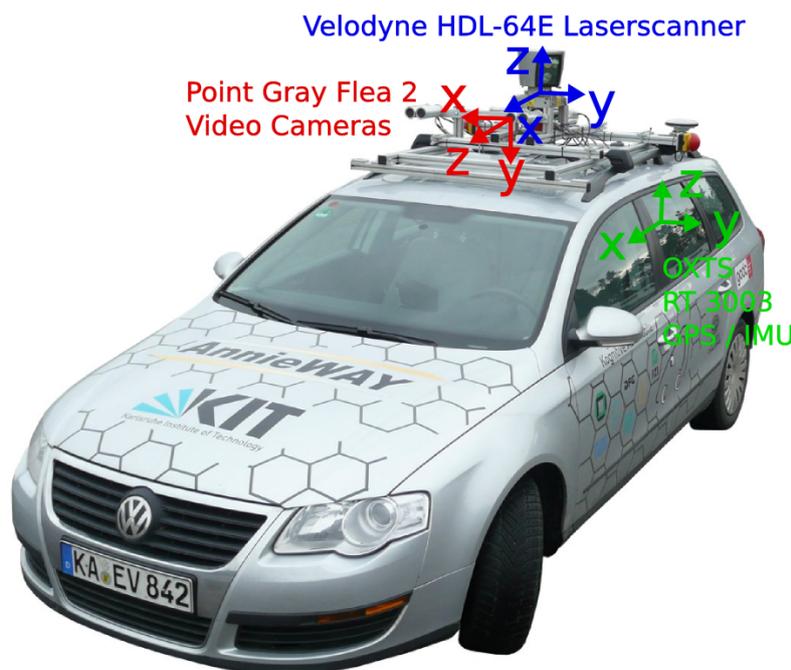


Figura 2.3: Vehículo de recogida de datos del KITTI.

- **Calidad de información:** Las medidas que otorga el escáner láser son más exactas que las extraídas por visión estéreo.
- **Coste computacional:** Se requiere un algoritmo de correspondencia y cálculo matricial a partir del par de imágenes obtenidas por visión para extraer la nube de puntos tridimensional, mientras que el escáner láser ofrece una información más directa.

A continuación mostramos dos algoritmos desarrollados previamente por otros investigadores, en uno de ellos se ha utilizado información extraída por un LIDAR y en el otro información a partir de visión estéreo.

2.4.1. Algoritmo de detección de borde con medición láser

Tomamos como ejemplo la referencia [9], en ésta se explica el funcionamiento de un algoritmo para detección de bordillos y rastreo de objetos utilizando un escáner 3D-LIDAR.

De la nube de puntos extraída se realiza una rejilla tridimensional de *voxels* cúbicos (unidades tridimensionales) para discretizar la información y facilitar su eficiente extracción, mejorando el tiempo de cómputo.

Se considera que un *voxel* contiene información de suelo si es un miembro del grupo vertical de *voxels* adyacentes no vacíos en una columna vertical.

Todos los puntos 3D almacenados en ese grupo de voxels son ajustados a un plano y los puntos contenidos en los *voxels* que cruza el plano serán los puntos de suelo. Los *voxels* de suelo pueden contener puntos de suelo o ser *voxels* vacíos, todos los que no sean de suelo serán considerados como potenciales obstáculos.

Los puntos de borde son diferentes de otros puntos de suelo en varios aspectos.

- La diferencia de elevación es mayor en la región del bordillo.
- La elevación cambia rápidamente en la dirección perpendicular al bordillo.
- La superficie del bordillo siempre apunta al centro de la vía.

En base a estas observaciones se propone seleccionar los puntos de bordillo a partir de los puntos de suelo usando la diferencia en elevación, el valor de gradiente de elevación y la orientación de superficie normal.

Primer criterio, diferencia de elevación: Si la diferencia de elevación de dos *voxel* vecinos difiere en un rango razonable se seleccionan los puntos.

Segundo criterio, gradiente de elevación: Se calcula la media de elevación de cada *voxel* y se asigna ese valor como elevación del *voxel*, la rejilla se convierte en un mapa de elevación 2D y se aplica un filtrado Gaussiano, entonces se aplica un operador Sobel, en las direcciones horizontal (G_x) y vertical (G_y). Debido al esparcimiento de los puntos no es apropiado aplicar el operador de forma directa, ya que varios *voxel* estarán vacíos, por tanto se necesita verificar la vecindad para aplicar el operador y se realiza para cada par de vecinos en el mapa de elevación, por ejemplo, para la matriz de *voxels* $I_{i,j}$, si $I_{i,j-1}$ ó $I_{i,j+1}$ no tienen valor de elevación, el

operador $G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$ se cambia a $\begin{bmatrix} -1 & 0 & +1 \\ 0 & 0 & 0 \\ -1 & 0 & +1 \end{bmatrix}$.

Es decir, sólo cuando un par de *voxels* son válidos computamos el gradiente utilizándolos, de la otra forma no se los considera. El operador *Mascara* a continuación representa el proceso de verificación al computar el gradiente.

$$G_x = I * Mascara * \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad (2.1)$$

Se seleccionan los puntos del *voxel* si su gradiente se encuentra en un cierto intervalo.

Tercer criterio, superficie normal: Para computar la normal de cada punto de suelo se utiliza el Método de Análisis de Componentes Principales y todos los puntos localizados en el mismo *voxel* son considerados como vecinos. Para cada punto vecino se construye su matriz de covarianza $Cov = \frac{1}{k} \sum_{i=1}^k (p_i - \bar{p})(p_i - \bar{p})^T$ donde k es el número de puntos localizados en el mismo *voxel* y \bar{p} es el centroide tridimensional de los vecinos. La normal será el autovector asociado al autovalor más pequeño. Se seleccionará un punto si su normal es perpendicular al coche.

Tras obtener los candidatos a bordillo se filtran los producidos por obstáculos en los que se observa la presencia de una gran discontinuidad vertical. Se rechazan estos falsos positivos utilizando una técnica de memoria a corto plazo: si un *voxel* dado ha sido clasificado como parte de la vía entonces cualquier detección de una discontinuidad en ese *voxel* no se trata como un bordillo cuando el *voxel* continúa en el rango del sensor. Debido al pequeño movimiento entre sucesivos *frames* o cuadros, podemos combinar los puntos de bordillos de varios cuadros anteriores con el actual, a la imagen resultante se le llama mapa de evidencia de bordillo.

Finalmente para el mapa de evidencia del bordillo se procede a realizar un ajuste parabólico para modelar los bordillos detectados. En cada fila del mapa se seleccionan dos *voxels* con la máxima puntuación de evidencia como los puntos de cresta, se toman dos, uno en un lado del coche y el otro al otro. Éstos tienden a ser falsos positivos, así que se modela el bordillo como

parabólico y se usa el algoritmo RANSAC para eliminar los falsos positivos que no concuerdan con el modelo parabólico.

No se expondrá en mayor detalle el ajuste parabólico. A continuación se muestran los resultados (ver figura 2.4)

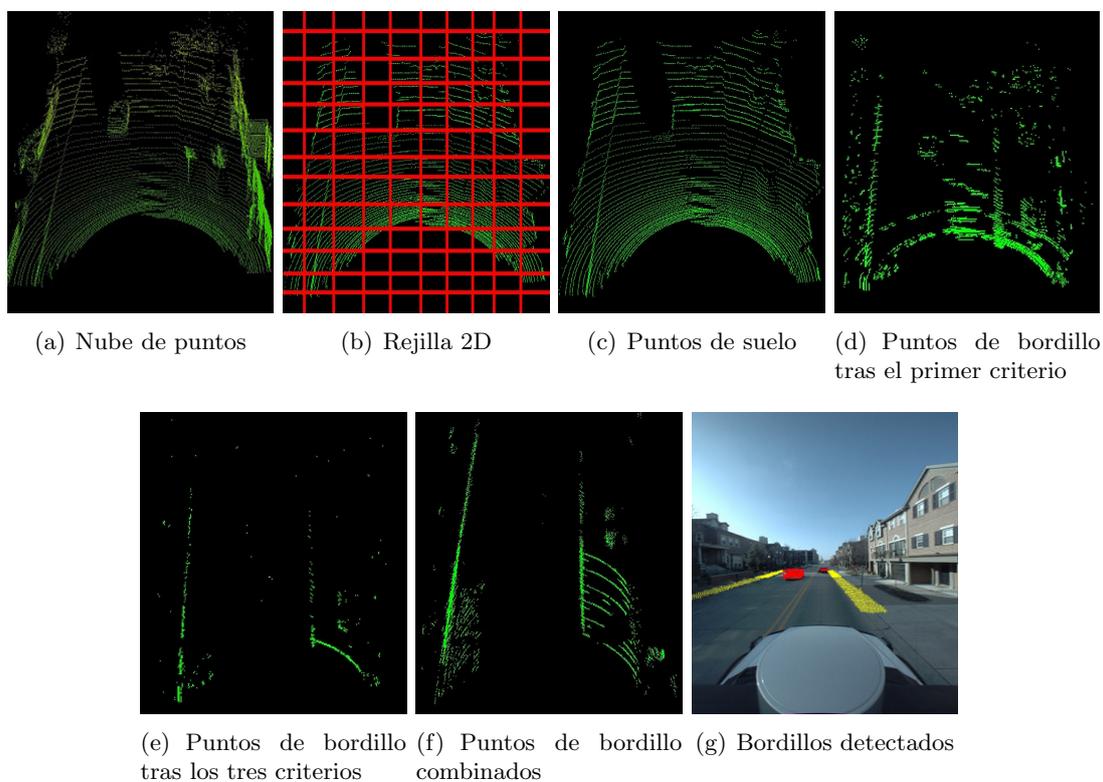


Figura 2.4: Resultado del algoritmo con láser

2.4.2. Algoritmo de detección de bordillo con visión estéreo

El algoritmo con visión estéreo propuesto en [10] es robusto si se asumen los siguientes requisitos:

- Se reconstruyen puntos 3D alrededor del bordillo y no predomina el ruido.
- Se presenta una variación de altura significativa (gradiente de elevación) alrededor del bordillo, mínimo 5 cm, y la variación de altura es rápida, como una función escalón.

El algoritmo trabaja siguiendo los siguientes pasos:

1. Construcción del Mapa de Elevación Digital (DEM).
2. Detección de los bordes en el DEM: se detectan puntos con una variación específica de altura.
3. Filtrado temporal de los bordes DEM: sólo los puntos de bordillo estáticos en relación a la vía son validados. Se propone un mapa de persistencia multi-cuadro.
4. Extracción de bordillos rectos y curvos del acumulador de Hough.

Construcción del Mapa de Elevación Digital (DEM)

A través de la nube de puntos extraída desde el estéreo se toman unos límites, una ROI (Región de interés) en tres intervalos en los ejes de la nube de puntos para generar a partir de ahí el DEM.

En la matriz resultante, a lo largo de las filas tenemos la profundidad de la nube de puntos y a lo largo de las columnas el ancho considerado visto desde las cámaras, incluyendo todos los puntos cuya elevación esté en el rango deseado de la ROI.

Las alturas son trasladadas y escaladas para una mejor visualización del DEM, pero las medidas originales son las que se usan en el procesado (ver figura 2.5).

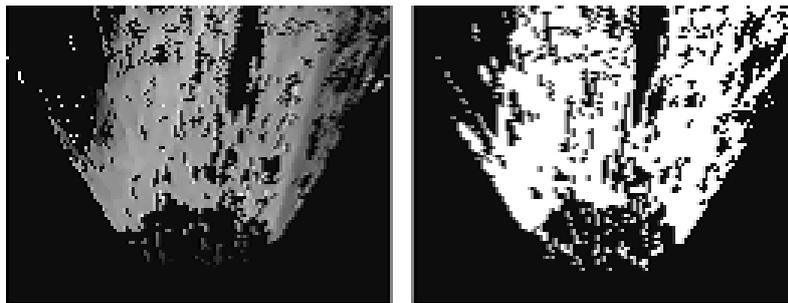


Figura 2.5: A la izquierda el DEM, más oscuro significa más alto, a la derecha el mapa de validez, que enmascara los puntos con información 3D.

Detección de bordes DEM

La mayor característica de un punto perteneciente a un bordillo es que su elevación varía bruscamente alrededor del mismo, se usa el método Canny para detectar los bordes, sólo los que poseen una variación equivalente a cierto rango de alturas son considerados y detectados.

Algunos falsos positivos se reconstruyen en la superficie de la vía causando bordes falsos. Este ruido aparece debido a la baja precisión de la reconstrucción de la superficie de la vía.

Este hecho puede dificultar el proceso de extracción de bordillos pues el ruido tiene variaciones de altura similares a bordillos pequeños. Esta es la principal razón por la cual a lo sumo se detectan dos bordillos, para evitar falsos positivos.

Filtrado temporal de los bordes DEM basados en persistencia multi-cuadro

El filtrado se basa en las siguientes consideraciones:

- Los bordillos son estáticos en relación a la superficie de la carretera: si la detección los puntos de bordillo del cuadro actual deberían solaparse a los del anterior.
- Los falsos positivos debidos al ruido deberían tener diferentes localizaciones entre cuadros consecutivos.

Esto se puede realizar conociendo la traslación y la rotación entre dos cuadros consecutivos, la cual se puede lograr a través de sensado, y los cambios de referencia entre un cuadro y otro calculados a partir de un modelo. La persistencia filtra los falsos positivos (Ver figura??). Otra ventaja del filtrado es que también elimina elementos dinámicos de la escena como peatones y obstáculos.

Extracción de bordillos del acumulador de Hough

Se construye un acumulador de Hough, los máximos locales del acumulador son líneas relevantes que pueden contener segmentos de bordillo. Sin embargo el acumulador no provee información acerca de cómo se reparten los puntos alrededor de las líneas relevantes. Por tanto se estudian manualmente para comprobar si son ruido o puntos de bordillo. Cada punto de la línea debe ser comprobado y comprobarse que cumpla las siguientes características:

- Su variación de altura se encuentra en un intervalo específico.
- La variación de su altura es positiva desde el lado de la vía hasta el lado opuesto.

Se utiliza una máscara circular de cierto tamaño sobre cada punto de la línea, la diferencia de las medianas de los puntos del lado del coche y de los contrarios sirve como la variación de altura para el punto y ha de ser positiva (Ver figura 2.6), combinados estos cálculos en los puntos forman el perfil de variación de alturas.

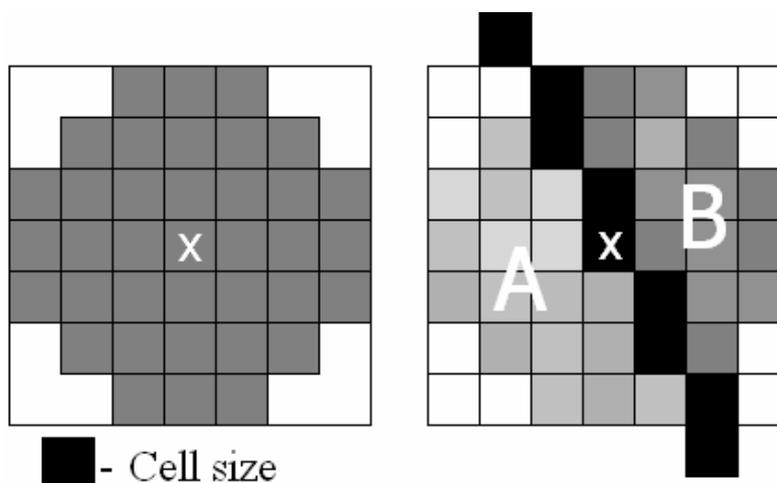


Figura 2.6: A la izquierda la máscara circular y a la derecha un ejemplo con datos reales, los píxeles de la línea están dibujados en negro.

Extracción de cadenas de segmento de los bordillos

Para modelar bordillos curvos se combinan en cadena segmentos de bordillo identificados a partir del acumulador y los criterios y extraídos de la siguiente forma:

1. Seleccionar el máximo global del acumulador.
 - a) Si el máximo supera un umbral:
 - i) Computar el perfil de variación de altura.
 - ii) Extraer el segmento de bordillo dominante representado por el intervalo continuo de más largo de puntos de bordillo del perfil de variación de altura (debe ser de al menos el número de puntos igual al umbral) Añadir recursivamente segmentos de bordillo a cada final del segmento dominante.
 - iii) Computar todos los pares de parámetros extraídos del acumulador de cada línea para cada punto válido de la cadena de segmentos de bordillo detectada y sustraer un voto del correspondiente en el acumulador.
 - b) Si no, parar.

2. Repetir el paso 1.

Como resultado podemos observar el siguiente ejemplo (Ver figura 2.7),

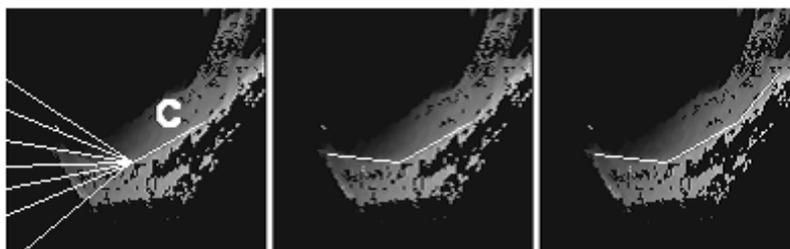


Figura 2.7: Un bordillo curvo se aproxima a una serie de segmentos de bordillo encadenados.

Después se desreferencia el bordillo del DEM y se proyecta en la imagen izquierda del estéreo para representarse (Ver figura 2.8)

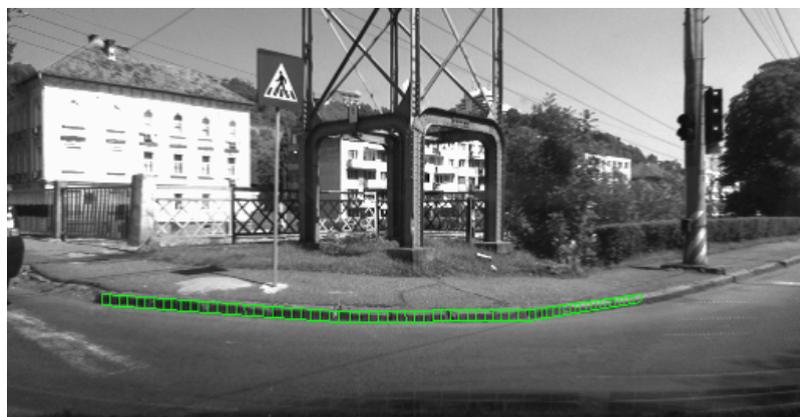


Figura 2.8: Resultado de la detección proyectado.

2.5. Objetivos

El objetivo del trabajo es mostrar una serie de resultados satisfactorios detectando bordillos en su parte no visible, hecho concreto que hasta ahora no se ha mostrado como resultado. En las fases de desarrollo del algoritmo se pueden establecer los siguientes subobjetivos:

1. Formar un mapa de elevación con información densa a través de la información tridimensional obtenida de la reconstrucción estéreo y los parámetros y la captura de una de las cámaras.
2. Tratar la información del mapa de elevación para facilitar la detección en la medida de lo posible
3. Idear un algoritmo que trabaje de forma innovadora para trazar el recorrido del bordillo en su parte no visible.

Capítulo 3

Formación de la perspectiva Vista de Pájaro

3.1. Parámetros de una cámara

En el trabajo realizado, es importante conocer los parámetros de una cámara, bien sea por la reconstrucción estéreo y los cambios de perspectiva, será de especial importancia la homografía vista de pájaro, la cual abate todo lo visto por la cámara en el plano horizontal.

Estos son parámetros intrínsecos o internos y extrínsecos o externos.

3.1.1. Parámetros intrínsecos

Los parámetros intrínsecos o internos dependen de la propia cámara y no de su posición. Existen cinco parámetros intrínsecos pero en la práctica solo utilizamos cuatro de ellos, estos son:

- $f_x = f/dx$, distancia focal en píxeles respecto del eje x.
- $f_y = f/dy$, distancia focal en píxeles respecto del eje y.
- (u_0, v_0) , coordenadas en píxeles del centro óptico.

Donde dx y dy son cambios de unidades (mm/px) en los ejes x e y respectivamente. Juntos forman la matriz de parámetros intrínsecos

$$K = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

3.1.2. Parámetros extrínsecos

Los parámetros extrínsecos no dependen de la cámara utilizada sino de la posición de esta misma respecto a la referencia del mundo que se esté utilizando.

Existen un total de 12 parámetros extrínsecos, que representan la rotación y traslación de la cámara respecto de la referencia origen, nueve para la matriz de rotación, de r_{11} a r_{33} , y tres

para definir la traslación t_x, t_y y t_z . Juntos forman la matriz de parámetros extrínsecos.

$$M = \begin{bmatrix} R_{3 \times 3} & T_{3 \times 1} \\ 0 & 1 \end{bmatrix} \quad (3.2)$$

Combinándolas se transforman las coordenadas mundo (X, Y, Z) en coordenadas pixélicas (u, v)

$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (3.3)$$

3.2. Formación de la vista de pajarero

El DEM que se ha utilizado comúnmente en el campo de la visión en entornos urbanos parte de que en el bordillo la información es densa, pero nuestro objetivo no se ve beneficiado por ese hecho ya que al no ser visto el bordillo no tenemos información directa de él, pero si del suelo superior a la vía, anterior a donde se encuentra y de la propia vía más allá del bordillo.

El mapa de elevación común hace que se pierda gran cantidad de información en esa zona, dificultando la identificación además de que la baja densidad de puntos otorga pocos puntos para poderlos identificar como puntos de bordillo, es por tanto que se propone la siguiente alternativa: un mapa de elevación abatido, basado en la homografía de vista de pájaro.

A partir de una sola imagen sin distorsión, con los parámetros de su cámara y desconociendo la información tridimensional que parte de la reconstrucción estereoscópica se puede formar una homografía de vista de pájaro (*bird view*) (Ver figura 3.1).

Esta transformación toma la falsa suposición de que todos los píxeles de la imagen poseen una altura cero y por tanto los proyecta como si los estuviera abatiendo al plano horizontal, en la sección de resultados se puede apreciar claramente este efecto (Ver figura ??).

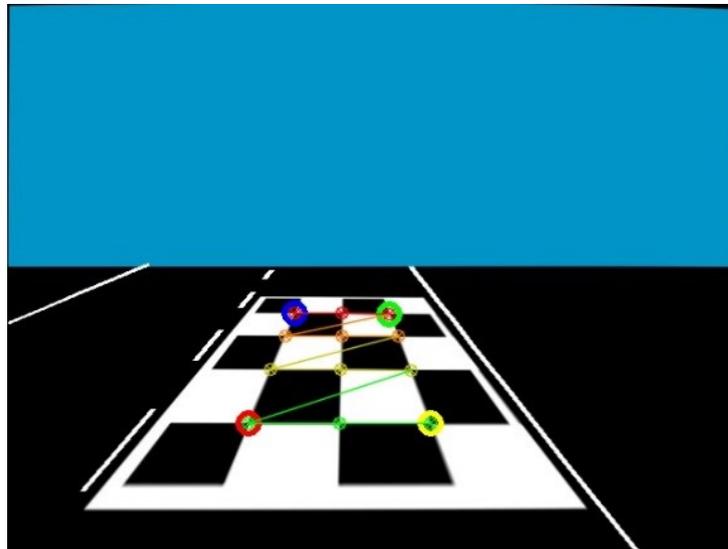
Para realizar la homografía necesitamos los parámetros extrínsecos e intrínsecos de la cámara además de unos parámetros de entrada para dimensionarla, que son: Profundidad d , que marca la distancia total a la que llegará la homografía de frente, anchura x , y píxeles por metro ppm , que marcará la resolución de la homografía y que junto a d y x condicionará el número de filas R y columnas C de la homografía.

$$R = d * ppmC = x * ppm \quad (3.4)$$

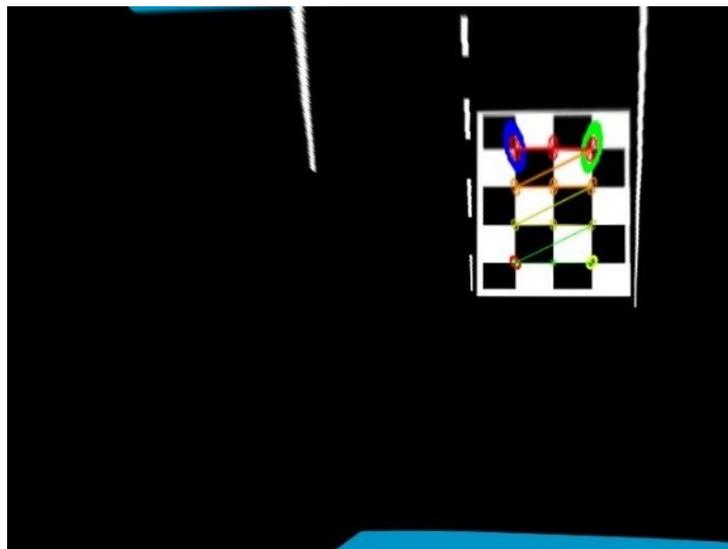
El tamaño de la homografía H condicionará en parte el tiempo de cómputo de los algoritmos utilizados sobre la misma, por tanto es recomendable no utilizar una resolución ppm excesiva y usar unos dimensionamientos que se ajusten al uso que se le dará.

Como se explicó anteriormente a través de los parámetros extrínsecos e intrínsecos se transforman las coordenadas reales en pixélicas, pasando de (X, Y, Z) a (u, v) . Si realizamos un barrido en (X, Y) en los límites que establecemos en la vista de pájaro y aplicamos la transformación, obtendremos las coordenadas (u, v) de la imagen y consultando su información de color o nivel de gris en dicha coordenada la llevamos al píxel correspondiente de la imagen destino.

Para la imagen fuente $S_{v,u}$, con imagen destino $D_{r,c}$, con un largo y ancho específicos a incluir en la homografía y una resolución ppm píxeles por metro, recorreremos en X y en Y



(a) Antes



(b) Después

Figura 3.1: Transformación a vista de pájaro

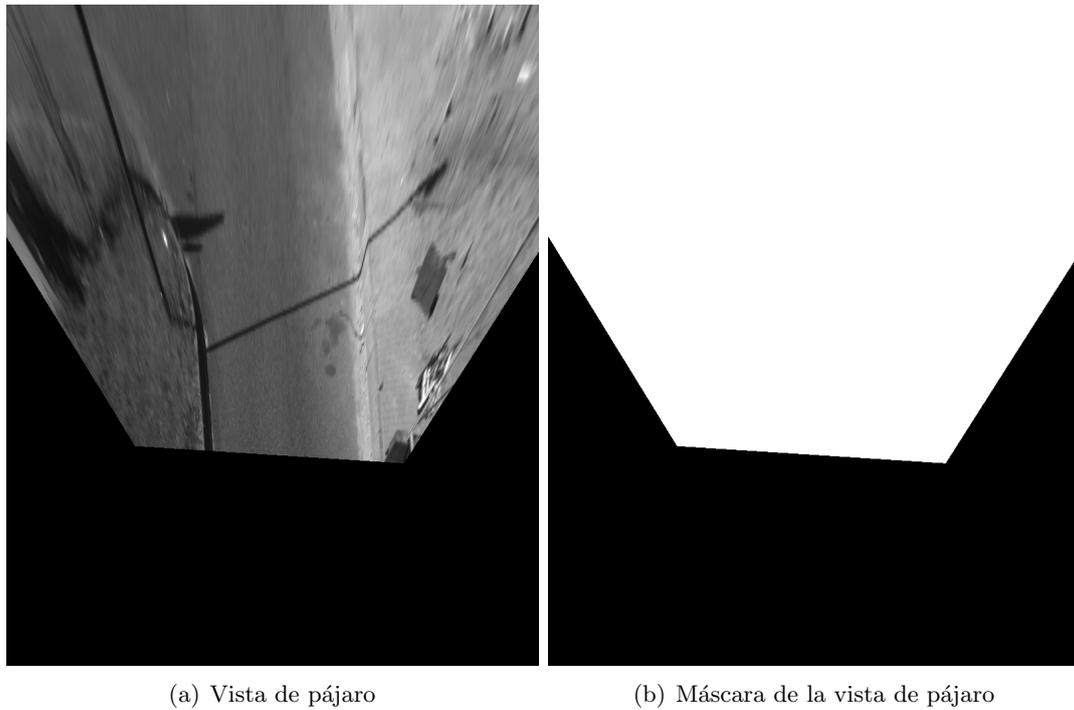


Figura 3.2: Ejemplo de una vista de pájaro utilizada y su máscara

el intervalo correspondiente a los límites especificados. Para cada par de (x, y) aplicamos la siguiente transformación:

$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} \quad (3.5)$$

Para cada par de (u, v) obtenido, se toma el valor $S_{u,v}$ si está dentro de rango en las dimensiones de la imagen fuente y se deposita en $D_{r,c}$, donde $r = R - x * ppm$ y $c = C/2 + y * ppm$.

Al formarse la vista de pájaro siempre habrá una zona vacía, en negro, esta zona la ocuparía el lugar que está por debajo del campo de visión de la cámara. Esta zona de la vista de pájaro donde no se ha depositado información se distinguirá más adelante porque no entra en la máscara de la vista de pájaro.

Vemos el ejemplo de una vista de pájaro en la figura3.2

Capítulo 4

Mapa de elevación abatido

4.1. Análisis previo

Típicamente los algoritmos que utilizan el mapa de elevación forman éste directamente a través de la información de la nube de puntos obtenida de la reconstrucción estereoscópica, consiguiendo una proyección directa de la nube sobre el plano que forma el suelo. Estos mapas poseen una densidad de información bastante superior a los obtenidos por tecnologías láser, pero aun así no conocemos información directa de los puntos que deseamos detectar, los del bordillo oculto.

Para obtener una mayor densidad de información alrededor de la zona a estudiar se ha decidido no tomar directamente la información de la nube de puntos proyectándola en el plano del suelo sino obtener la información tridimensional de cada punto representado en la propia vista de pájaro.

La nube de puntos formada a partir de la reconstrucción estereoscópica surge del algoritmo de correspondencia. Para cada punto visto en una de las dos imágenes, izquierda o derecha se obtiene un valor tridimensional, o, si no hay correspondencia entre ambas imágenes, un punto sin disparidad. Si representamos en una de las dos imágenes la información tridimensional de cada punto en vez de su color o nivel de gris, obtenemos una imagen de información tridimensional, además, con un cambio de escala podemos representar en el espacio BGR dicha información. (Ver figura 4.1)

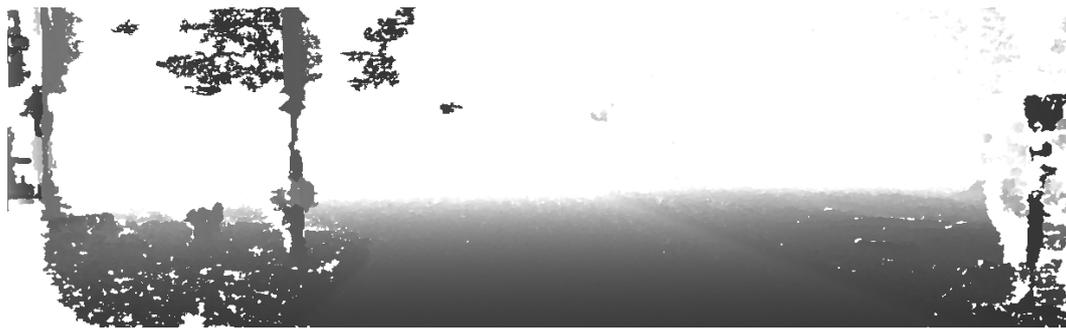
Si tomamos el canal correspondiente a la elevación y formamos una vista de pájaro con el resultado obtendremos un mapa de elevaciones abatido. Sin embargo la reconstrucción no tiene en cuenta la inclinación de la vía y por tanto la coordenada Z no está correctamente referenciada, siendo de esta forma inválida.

4.2. Formación del mapa de elevación abatido

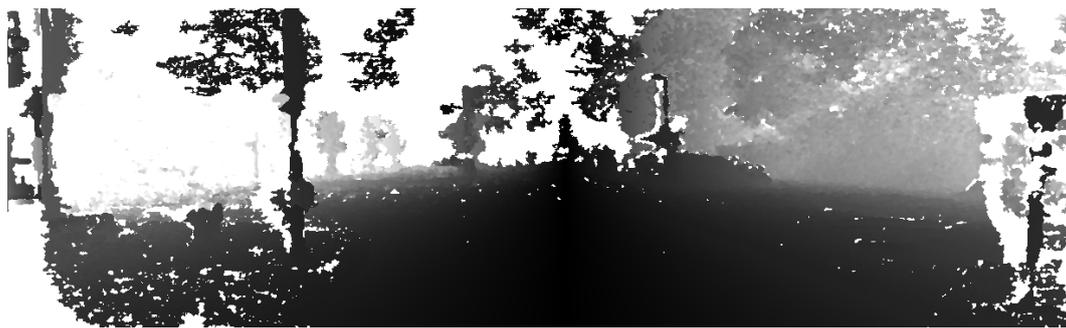
4.2.1. Extracción del plano principal

Para solventar el problema de la referenciación de alturas hemos de encontrar el plano del suelo en la nube de puntos tridimensional.

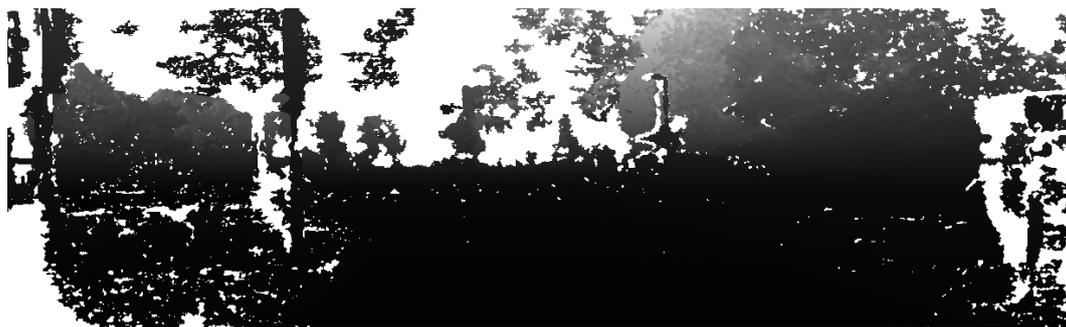
En primer lugar se han de computar las normales para la nube de puntos actual y después se siguen los siguientes pasos:



(a) Componente x



(b) Componente y



(c) Componente z



(d) En BGR

Figura 4.1: Representación xyz en escala de grises, los puntos sin disparidad se saturan

- Se eliminan los obstáculos en un rango entre cierta distancia y se colocan en una nube temporal.
- Se eliminan los puntos más alejados al rango del vehículo y se almacenan en la nube temporal.
- Con los puntos restantes se ajusta un plano, este corresponde aproximadamente con el plano del suelo.

Del plano principal se extraen sus coeficientes y con ellos y la posición tridimensional de los puntos Z podemos establecer la elevación referenciada al plano del suelo h , que no deja de ser la distancia de un punto a un plano.

$$h = \frac{c_0X + c_1Y + c_2Z + c_3}{\sqrt{c_0^2 + c_1^2 + c_2^2}} \quad (4.1)$$

4.2.2. Obtención del mapa de normales

Para la extracción del plano principal se tuvieron que calcular las normales previamente, aprovechando esta necesidad y conociendo la correspondencia de cada punto de la nube con la de su imagen de la cámara, se puede formar una imagen o matriz de normales, también podemos escalar esta misma imagen para obtener una representación de cada componente (x, y, z) de las normal en cada punto en el espacio BGR.

De la misma forma que se realiza la vista de pájaro se puede realizar la homografía cambiando la imagen de color o valores de gris por la de normales, obteniendo una vista de pájaro de normales o mapa de normales.

4.2.3. Colocación de elevaciones

Con la imagen con información tridimensional (X, Y, Z) y los coeficientes del plano principal realizamos el barrido en (X, Y) en el intervalo especificado por los límites de la vista de pájaro aplicando la transformación 3.5, esta vez tomamos toda la información tridimensional y calculamos la distancia al plano principal dada por la ecuación de distancia de un punto al plano (4.1), depositamos esta distancia en la imagen destino $D_{r,c}$, donde $r = R - x * ppm$ y $c = C/2 + y * ppm$.

Esta acción anterior no se realiza directamente, se estudia antes la información tridimensional del punto para tener en cuenta si conviene depositar su elevación o no. Para la representación gráfica se tomarán una serie de condiciones.

A la hora de colocar las elevaciones es de vital importancia que se tome un rango razonable de puntos para que toda la superficie superior al bordillo sea incluida, ya que la búsqueda del bordillo no visible se basará en las superficies superior y la de más allá del bordillo.

A partir de este momento distinguiremos la matriz de elevaciones y la imagen de elevaciones. La primera incluirá la información en coma flotante con la elevación en metros, la segunda el valor de gris utilizado para su representación.

Se tomarán las siguientes condiciones para depositar una elevación tanto en la matriz como en la imagen:

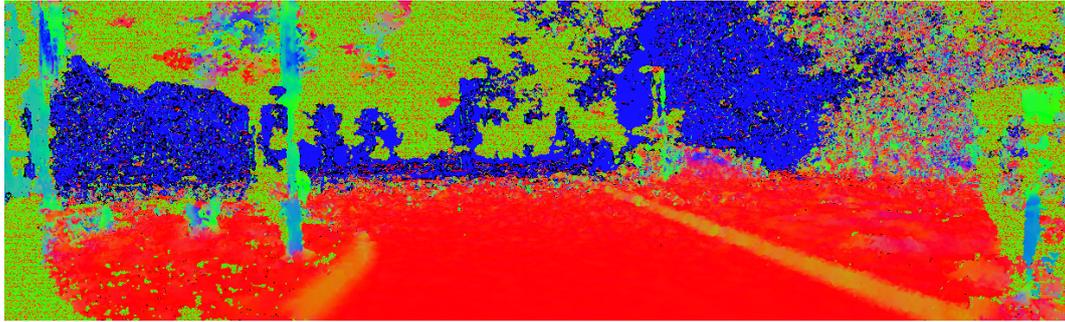
- **Puntos sin disparidad:** Poseerán una altura errónea y una profundidad muy elevada tras la reconstrucción estéreo, estos no se incluirán, ya que son falta de información.



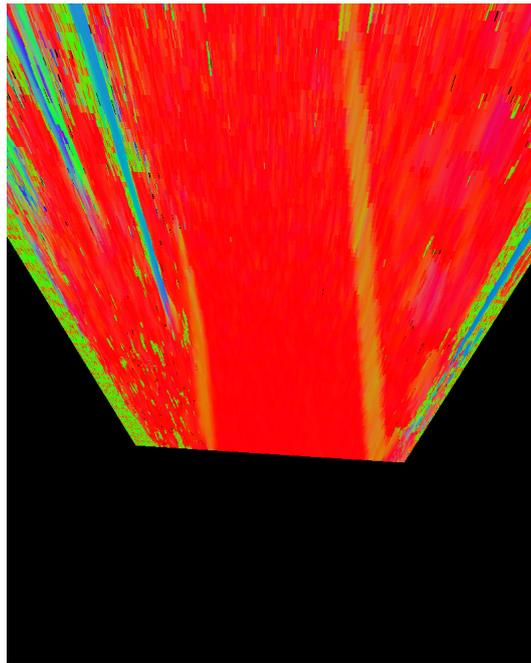
(a) Componente x

(b) Componente y

(c) Componente z



(d) Representación BGR



(e) BGR a vista de pájaro

Figura 4.2: Representación de normales

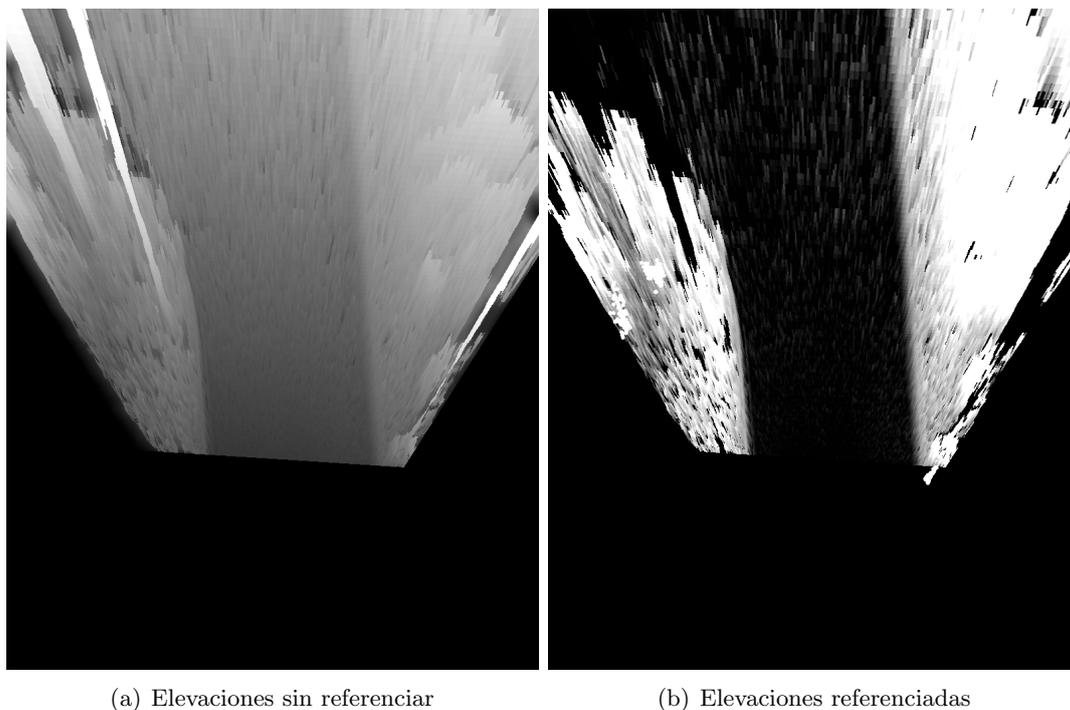


Figura 4.3: Efecto del plano principal

- **Elevación superior a un umbral:** La elevación se rebaja al nivel del umbral, este umbral se escoge para que incluya elevaciones de hasta 20cm, alturas superiores al umbral e inferiores al doble del mismo serán saturadas al umbral.
 - **Elevación superior al doble del umbral:** La elevación no se incluye, al ser muy elevados serían considerados obstáculos.
 - **Elevación inferior a 0:** Estos puntos aparecen por la propia curvatura del suelo o la inexactitud del plano principal, en cualquier caso serán puestos a la elevación mínima.
- **Rango de elevación razonable:** Los puntos fuera de rango se incluirán de forma modificada o no se incluirán.
 - **Puntos en superficie vertical:** Utilizando la información de las normales, aquellos con baja componente z pueden ser considerados puntos en una superficie vertical, este criterio no influirá en que se incluya la elevación tal cual o no, pero se considerarán posteriormente.

El efecto de la referenciación al plano principal se puede apreciar en la figura 4.2.3

4.3. Enmascaramiento de regiones

Como se puede observar hay ciertas zonas de la matriz e imagen que han sido modificadas para favorecer la detección. Todos estos elementos forman diferentes máscaras a medida que se van colocando siguiendo los criterios de no disparidad, elevación superior y puntos en superficie vertical.

En el caso de las zonas con gran elevación, como las que forman farolas, postes o similares al abatirse sobre el plano $Z = 0$, formarán información falsa sobre la matriz e imagen. Este mal necesario se puede eliminar en gran medida con las máscaras que se han ido formando.

Las máscaras formadas son las siguientes:

- **Máscara de no disparidad:** Incluidos todos los elementos de la matriz que no poseían correspondencia en el estéreo y por tanto no se ha podido obtener su información tridimensional.
- **Máscara de gran elevación:** Incluidos todos los elementos que superaban el doble del umbral de elevación y que no se incluyeron en la matriz.
- **Máscara de verticales:** Incluidos todos los elementos que poseían una componente normal z inferior a un umbral y por tanto pertenecen a una superficie vertical.
- **Máscara de obstáculos:** Combinación lógica AND de las máscaras de gran elevación y verticales.

Las tres primeras se aprecian en la figura 4.3, la última se obtiene en forma de puntos y puede verse en el capítulo de resultados.

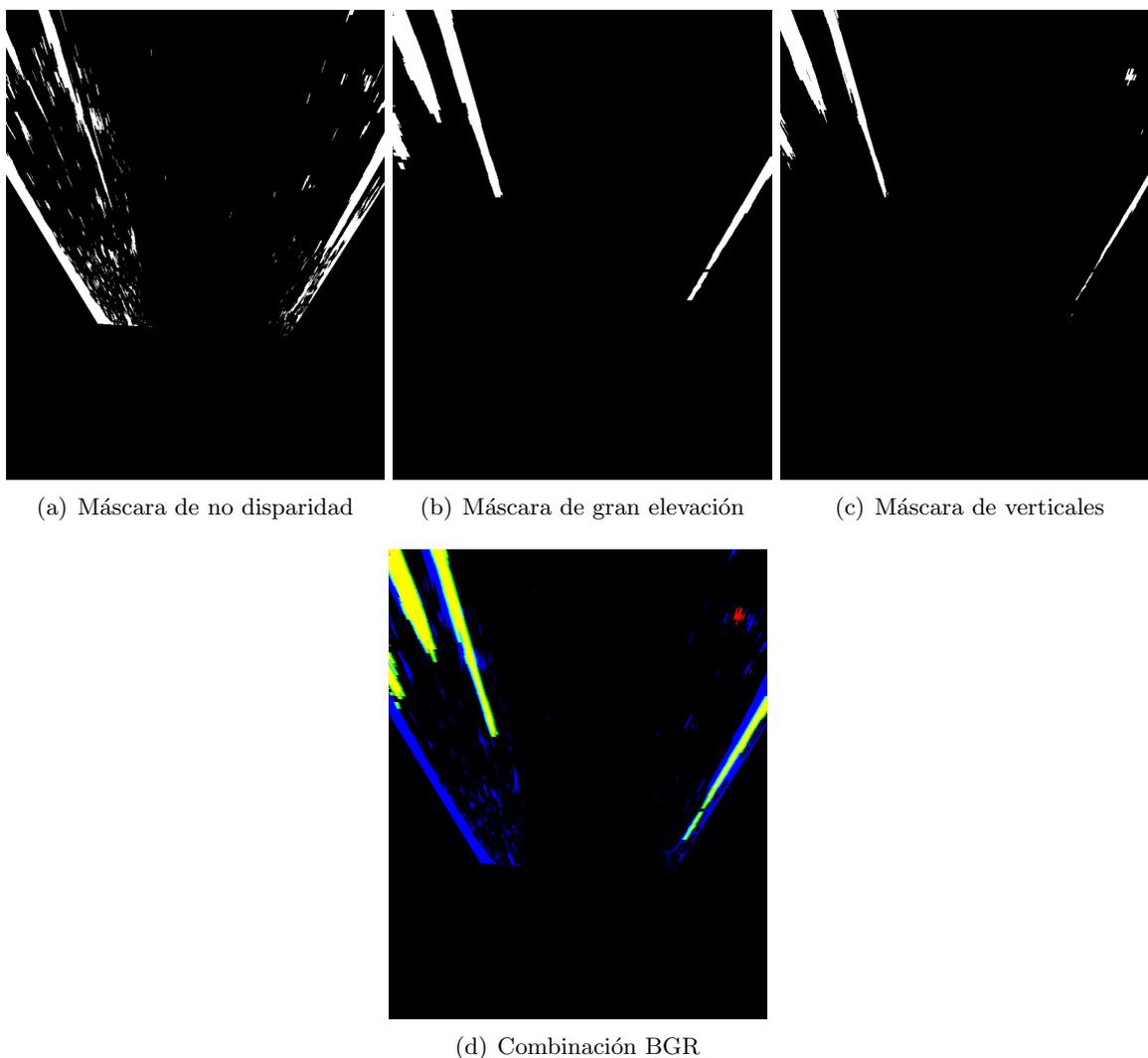


Figura 4.4: Enmascaramiento de regiones

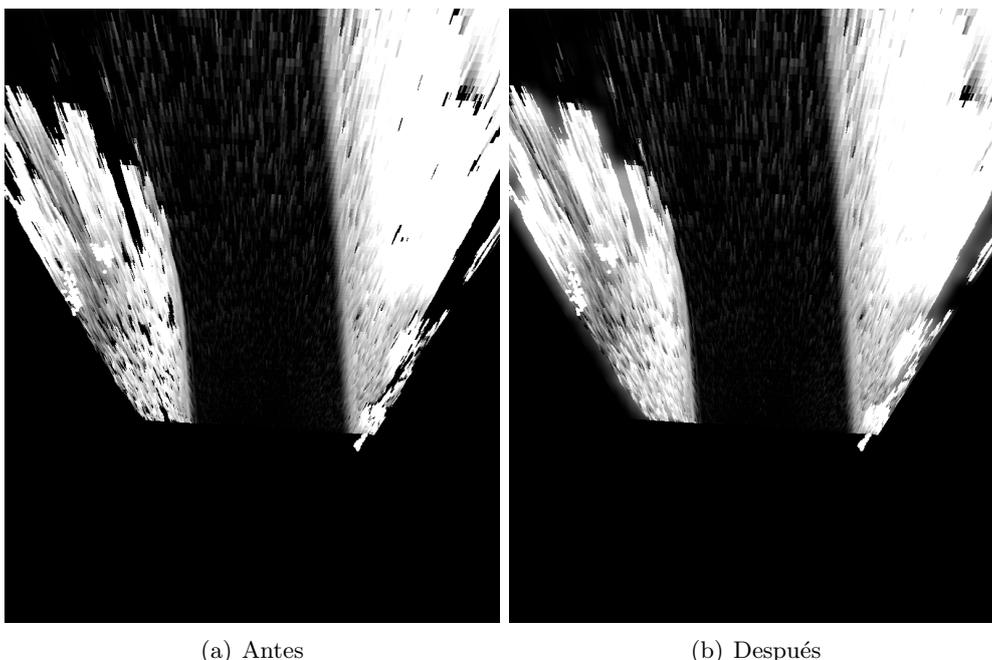


Figura 4.5: Rellenado parcial de información en objetos verticales

4.4. Inclusión de obstáculos

Dentro de la base de datos utilizada la cantidad de secuencias con un bordillo de las características requeridas es baja, y en una buena cantidad encontramos la presencia de algún obstáculo cerca o encima del mismo, como un vehículo estacionado.

Dada la peculiaridad de la presencia del obstáculo, el algoritmo utilizado actúa de forma diferente. Por tanto se ha decidido incluir puntos de obstáculo en el mapa de elevación abatido con el fin de combinar su información con la de la propia detección del bordillo e influir en ésta.

La detección de obstáculos está poco elaborada ya que no es el fin de este trabajo, los puntos de obstáculo son aquellos que se incluyeron en la máscara de obstáculos, éstos aparecen dispersos en el obstáculo al hallarlos, para solventar el problema cada punto se representa como un círculo con poco radio en el mapa de elevación.

4.5. Problema de los postes

Cuando el ojo humano ve un poste la información que desaparece tras él no es ningún inconveniente, el ser humano abstrae que tras el poste hay una continuidad de lo que está observando a izquierda y derecha. De forma similar se puede lograr reconstruir el espacio vacío que se deja al colocar las elevaciones en la matriz utilizando la máscara de verticales, además a las zonas sin disparidad se les puede aplicar el mismo criterio.

Para solventar en cierta medida estos problemas se realiza un filtro media sobre una copia de la matriz para añadirse posteriormente a las zonas con máscara de verticales y también a las que no tienen disparidad, Dando como resultado que esas zonas, si son estrechas, sean rellenadas por información estimada tal como se aprecia en la figura 4.5.

En este punto el mapa de elevación abatido y sus máscaras han sido completamente formados.

Capítulo 5

Algoritmo desechado: Ajuste de elipses

Dada la novedad del problema que se presenta en este trabajo, no se ha podido dar con una solución adecuada desde el principio y se han tenido que experimentar varias formas para encontrar una solución al problema, en un principio se comenzó a realizar el algoritmo bajo la suposición de que el bordillo oculto a encontrar poseía una forma de arco de circunferencia o en el peor de los casos elíptica.

Bajo esta suposición y utilizando el mapa de elevaciones abatido se ha intentado realizar una búsqueda de una elipse que encajara con el bordillo de forma aproximada para más adelante centrarse en el estudio de dicha zona, este método sólo ha dado resultados satisfactorios en pocos casos y para bordillos con dicha forma.

En primer lugar se realiza un barrido variando ciertos parámetros de un cuarto de elipse, buscando en ambos lados del vehículo, los parámetros se pueden apreciar en la figura 5.1

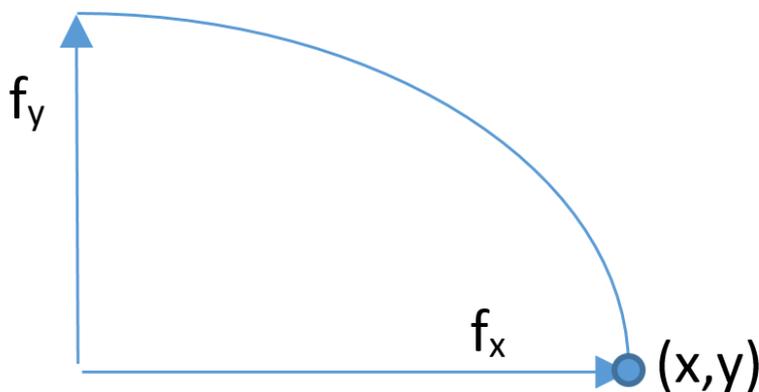


Figura 5.1: Parámetros utilizados para el barrido de elipses.

Estos parámetros son el punto donde inicia el arco y sus dos distancias focales. Para realizar el barrido es necesario encontrar un equilibrio entre velocidad y densidad, si se varían poco a poco los parámetros se tardará mucho en realizar el barrido y si se varían muy bruscamente es posible que ninguna elipse encaje con la posición del bordillo.

A la que vamos realizando el barrido cada elipse tiene que tomar información del mapa de

elevación abatido, para ello se forman dos máscaras, superior (M_S) e inferior (M_I), estas dos máscaras se forman a partir de otras tres, el relleno de la elipse original E_0 , el de otra más pequeña E_p y el de otra más grande E_g , estas dos máscaras se forman como se indica en 5.1.

$$M_S = E_0 - E_p \quad M_I = E_G - E_0 \quad (5.1)$$

Cada una de estas máscaras recoge las alturas del mapa de elevación abatido, obteniendo para cada elipse un vector de alturas superiores y otro de alturas inferiores, con las cuales se obtienen medidas estadísticas como la media, la mediana y la desviación típica.

En un principio se utilizaron la diferencia de medias y la diferencia de medianas como criterio para escoger la elipse mejor ajustada, en casos sencillos los resultados fueron satisfactorios pero en otros se encontraban lejos de serlo, en la figura 5.2.

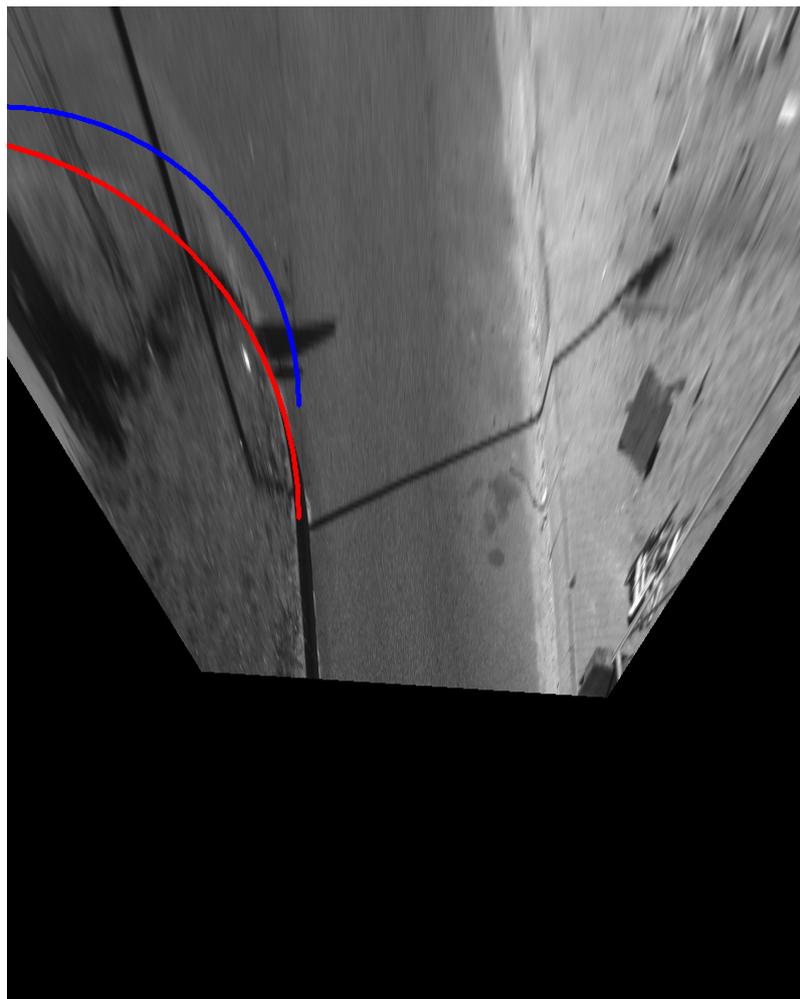


Figura 5.2: Resultado en la vista de pájaro del barrido de elipses. En rojo la que mejor se ajusta según diferencia de medias, en azul según diferencia de medianas.

Para usar el precepto de que el algoritmo se utiliza tras la localización de bordillos visibles, se indicaba manualmente el punto donde pasa a ser no visible para que sólo haga un barrido paramétrico de las distancias focales.

Los resultados que ofrecían este método eran muy inconsistentes y el método era poco elaborado, se descartó rápidamente.

Capítulo 6

Algoritmo de gusano

6.1. Idea general

Bajo la condición de una detección previa de un bordillo es donde el algoritmo desarrollado comienza a realizar su función. Bajo esta condición desaparece la necesidad de que el mismo ubique en la matriz el candidato y partamos de una posición conocida para llegar a la detección del bordillo oculto.

El algoritmo es capaz de seguir bordillos rectos y visibles a la par que los ocultos, por tanto su aplicación en ausencia de una parte oculta no supondría un problema. Del bordillo visible previamente detectado por otro algoritmo se debería obtener información tal como el punto más cercano al coche y su orientación inicial.

Se trata de un algoritmo iterativo, al que se ha bautizado como gusano, a cada paso del algoritmo el gusano avanza por el bordillo en el mapa de elevación abatido que se describe en el capítulo [?] y va depositando puntos que lo describen según un avance por paso definido, el conjunto de puntos que el algoritmo devuelve pueden utilizarse para modelar el resultado.

6.2. Criterio de bordillo

Para el uso del algoritmo se formula un criterio ante el cual se establece si estamos ante la presencia de un bordillo o no, según el contenido del mapa de elevación a un lado del bordillo contamos con la presencia de gran número de puntos bajos en un lado y en el otro un gran número de puntos altos, bajo este criterio supongamos un conjunto de N_i elevaciones en un lado izquierdo E_i y otro en el lado derecho E_d , de N_d elevaciones. En primer lugar calculamos la media y desviación típica de la unión de ambos conjuntos \bar{E} y σ_E .

A continuación para cada conjunto se comprueban una a una las elevaciones de la izquierda e_i y las de la derecha e_d , aquellas que sean correctas pertenecerán al conjunto de elevaciones correctas E_c , que son aquellas que cumplen estas condiciones, según un parámetro constante $K < 1$ escogido.

$$e_i \in E_c \quad \text{si} \quad e_i > \bar{E} + K \cdot \sigma_E \quad e_d \in E_c \quad \text{si} \quad e_d < \bar{E} - K \cdot \sigma_E \quad (6.1)$$

en el caso de que el bordillo sea izquierdo.

$$e_i \in E_c \quad \text{si} \quad e_i < \bar{E} + K \cdot \sigma_E \quad e_d \in E_c \quad \text{si} \quad e_d > \bar{E} - K \cdot \sigma_E \quad (6.2)$$

en el caso de que el borde sea derecho.

El final de este criterio es no establecer un criterio utilizando umbrales de alturas, el ruido inherente del estéreo provoca que para este criterio, si ambos conjuntos pertenecen a superficies de similar altura, gran cantidad de puntos sean desechados por la adición de la desviación típica y otra gran cantidad por estar en la otra dirección de la media.

Por último el criterio de borde m se evalúa como

$$m = \frac{E_c}{N_i + N_d} \quad (6.3)$$

6.3. Nacimiento

6.3.1. A partir de borde visible

El gusano es una figura geométrica, en principio un rectángulo dividido longitudinalmente por la mitad formando una máscara izquierda y derecha. Según el lado en el que haya sido detectado el borde estas máscaras se intercambiarán de posición, estas dos máscaras son la máscara de altos y la máscara de bajos.

Tres parámetros definen la geometría del gusano: Largo L , ancho W y avance A . La parte central del gusano se ubica en el inicio del borde, si éste tiene información de ángulo debe rotarse para orientar la línea divisora de las máscaras de forma paralela al borde.

6.3.2. Ubicado automáticamente

Al no disponer de un borde detectado previamente en el desarrollo y pruebas del algoritmo se ha realizado una selección de capturas en las que estaban presentes bordes ocultos, soltando gusanos a ambos lados de la escena, para ubicar el borde hacemos uso de la matriz de normales formada a partir de la imagen de normales.

Una característica de los bordes visibles previos a los no visibles que se encuentran en la base de datos utilizada es que son paralelos a la dirección del vehículo, y poseen una componente normal característica, observando la representación de las componentes normales en los tres canales BGR de una imagen a vista de pájaro se observa un tono anaranjado y una zona densa en las zonas con borde visible.

Haciendo uso de este conocimiento convertimos los canales BGR en HSV , y con el rango de tono del naranja aislamos este color consiguiendo una imagen que marque puntos con posible borde, se puede ver un ejemplo en la figura 6.3.2.

Para identificar el lugar donde se encuentra el borde, iteramos un rectángulo a lo largo de los límites de la vista de pájaro que recoja los puntos de la imagen anterior, el lugar con mayor densidad de esos puntos será considerado el borde, ya que a priori es sabido que existe, pues el único objetivo de este paso es automatizar las pruebas del algoritmo.

Este mecanismo podría haberse obviado iterando el gusano en vertical por todo el límite de la máscara de la vista de pájaro y buscando el mejor resultado para borde pero a través del otro método se gana en velocidad y exactitud.

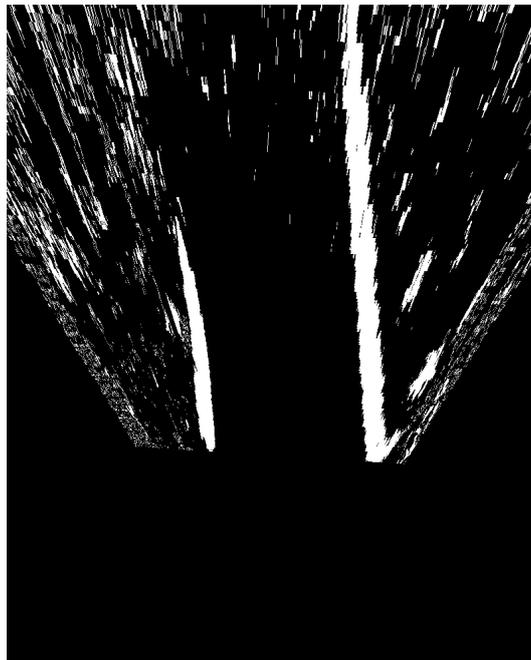


Figura 6.1: Zonas con componente normal que forma un color anaranjado al representarse en HSV las normales

6.3.3. Ajuste al bordillo

Al soltar el gusano sobre el bordillo previamente detectado o en el lugar indicado por la ubicación para la automatización podría darse un problema de posicionamiento por el abatimiento del mapa que requiera un ajuste fino para ubicarse justo en la posición del bordillo. El gusano iterará en las columnas adyacentes a izquierda y derecha un pequeño intervalo en la matriz para ver donde se cumple mejor el criterio de ubicación de bordillo, tomará el que de mejor resultado y comenzará a describir el bordillo desde ese punto.

6.4. Búsqueda

Para ir describiendo el bordillo el algoritmo itera y el gusano avanza el avance A hasta que da con el límite de la matriz, para ello el gusano busca a partir de su ángulo θ , desde un ángulo $\theta + \alpha$ hasta un ángulo $\theta + \omega$ siendo α un ángulo positivo en dirección hacia los bordes de la matriz, θ el ángulo que posee el gusano al haberse depositado en el punto en que está y ω positivo hacia el centro de la matriz. Es decir, si el bordillo está en el lado derecho, α será positivo en sentido horario y ω en sentido antihorario y en el caso del lado izquierdo de forma contraria como se puede observar en la figura 6.2.

Estos ángulos límite se basan en que los bordillos no deberían girar en un pequeño avance muchos grados, el ángulo ω no debe ser un ángulo ni grande ni pequeño ya que si el gusano se introduce hacia dentro del bordillo en una iteración no podría regresar hacia afuera.

Paso a paso la búsqueda del siguiente punto de bordillo seguiría los siguientes pasos:

1. Iteraciones desde el ángulo $\theta + \alpha$ hasta el ángulo $\theta + \omega$, en pequeños pasos, desde unos dos a cinco grados, recogiendo la correspondencia con el criterio de bordillo en cada iteración.

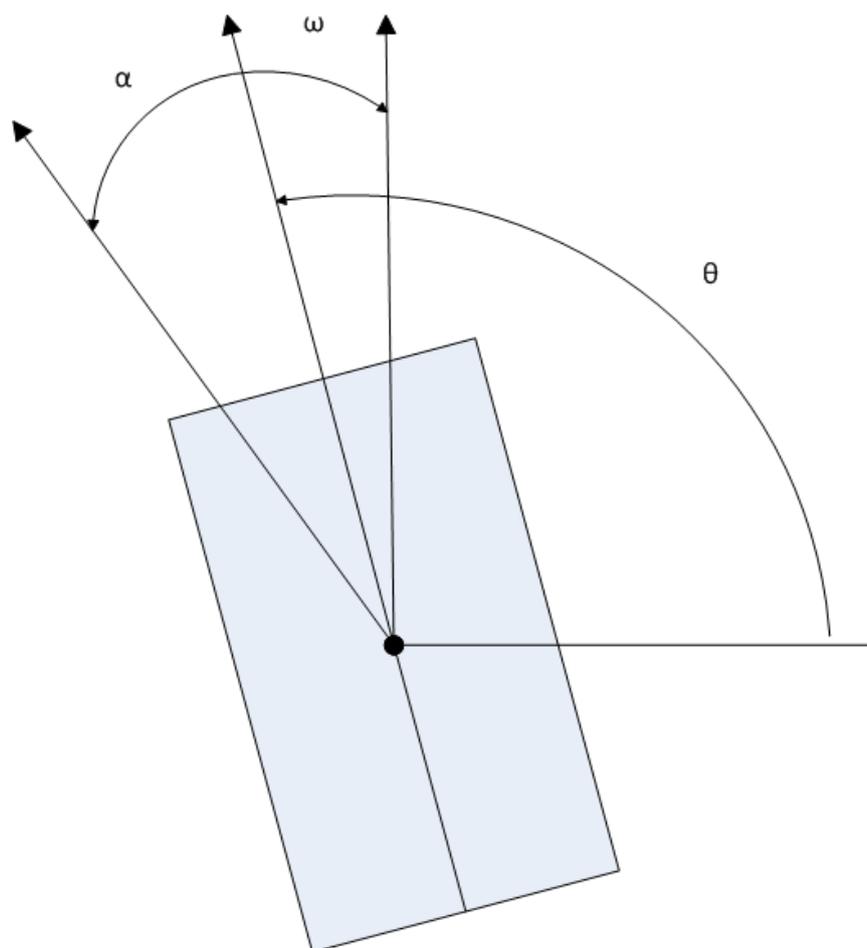


Figura 6.2: Ángulos de búsqueda del gusano en el lado izquierdo.

2. Posicionamiento del gusano en el ángulo con mayor correspondencia de bordillo
3. Avance en línea recta de la distancia de avance A hasta la nueva posición, pasando a ser su ángulo actual el nuevo ángulo $theta$ y la nueva posición un nuevo punto de bordillo.
4. Repetición de los pasos anteriores hasta que no se de con un criterio de bordillo suficientemente consistente, inferior a cierto umbral o hasta que se alcance un punto cercano a los límites de la matriz de vista de pájaro.

6.4.1. Geometría

A la hora de generar un gusano sobre la matriz de alturas es de importancia el ángulo con que se coloca y la posición en la que se queda, además de los puntos medios frontal y trasero que dividen su cuerpo en lado izquierdo y derecho.

Es de especial importancia conocer las coordenadas en la imagen para calcular las nuevas posiciones que tomará el gusano y aquellas en las que comprobará el criterio de bordillo, para ir avanzando por el bordillo se toman en cuenta los siguientes gusanos, definidos geoméricamente por cuatro parámetros:

- Largo del gusano L
- Ancho del gusano W
- Posición del gusano (x, y)
- Orientación del gusano θ

Teniendo en cuenta que las coordenadas (x, y) de la matriz avanzan positivamente hacia la derecha y hacia abajo, la posición final (x_f, y_f) del gusano al realizar un avance A viene dada por la expresión 6.4.

$$(x_f, y_f) = (x + A \sin(\theta), y - A \cos(\theta)); \quad (6.4)$$

Para colocar el rectángulo que representa el cuerpo del gusano se utiliza una función propia de las librerías *OpenCV* que crea un rectángulo rotado especificando centro, dimensiones y ángulo a rotar.

El gusano se divide en dos máscaras izquierda y derecha, de igual tamaño, se toman los puntos de las esquinas del gusano, accesibles fácilmente con las funciones. Los puntos medios entre los dos delanteros y los dos traseros forman dos máscaras con los que se encuentran a los lados.

Combinando la esquina de menor coordenada x y la de menor coordenada y obtenemos una coordenada (x, y) y con los de mayor otra, estas dos coordenadas definen un rectángulo no rotado que puede usarse como región de interés en la matriz para comprobar que puntos se encuentran dentro de cada máscara con ayuda de las librerías optimizando más el tiempo de procesamiento.

6.4.2. Criterio de certeza

Para la búsqueda del siguiente punto de bordillo se ha realizado un filtro de certeza consistente en una doble función gaussiana con un tramo constante, la cual decrece más lentamente en el lado del bordillo y más rápidamente en el de la vía, dada la probabilidad de que el bordillo no visible se torne hacia el lado opuesto de la vía, ya que de forma contraria sería visible y el algoritmo no entraría en juego.

La finalidad de este criterio es evitar que el gusano de giros bruscos a menos que se vaya a topar con un obstáculo reduciendo el criterio de bordillo en ángulos de búsqueda respecto al actual grandes.

Considerando el ángulo de búsqueda γ , que es el que posee el gusano en el lugar que está probando el criterio de bordillo y el ángulo actual del gusano θ , siendo el incremento del ángulo de búsqueda $(\gamma - \theta)$, la función está definida en tres trozos:

- El incremento del ángulo de búsqueda, si es en dirección al borde de la matriz (favorable), respecto al actual del gusano, supera un ángulo umbral favorable α .
- El incremento del ángulo de búsqueda, si es en dirección al centro de la matriz (desfavorable), respecto al actual del gusano, supera un ángulo umbral desfavorable ω
- El incremento del ángulo de búsqueda está entre los dos ángulos umbrales.

Estos ángulos umbral no son los mismos a los previamente detallados en la explicación de la búsqueda, la doble función gaussiana tiene dos desviaciones típicas, σ_f y σ_n , siendo $\sigma_f > \sigma_n$, es la siguiente:

$$\begin{aligned}
 & \text{factor} = \\
 & \frac{(\gamma - \theta) - \alpha^2}{e^{-2\sigma_f}} \quad \text{si } (\gamma - \alpha) > \alpha \quad \text{y } (\gamma - \theta) \text{ incremento hacia el borde} \\
 & 1 \quad \text{si } (\gamma - \alpha) < \omega \quad \text{y } (\gamma - \alpha) < \alpha
 \end{aligned} \tag{6.5}$$

$$\frac{(\gamma - \theta) - \omega^2}{e^{-2\sigma_n}} \quad \text{si } (\gamma - \omega) > \omega \quad \text{y } (\gamma - \theta) \text{ incremento hacia el centro}$$

La función toma la forma que se aprecia en la figura 6.3 siendo el eje x $(\gamma - \theta)$.

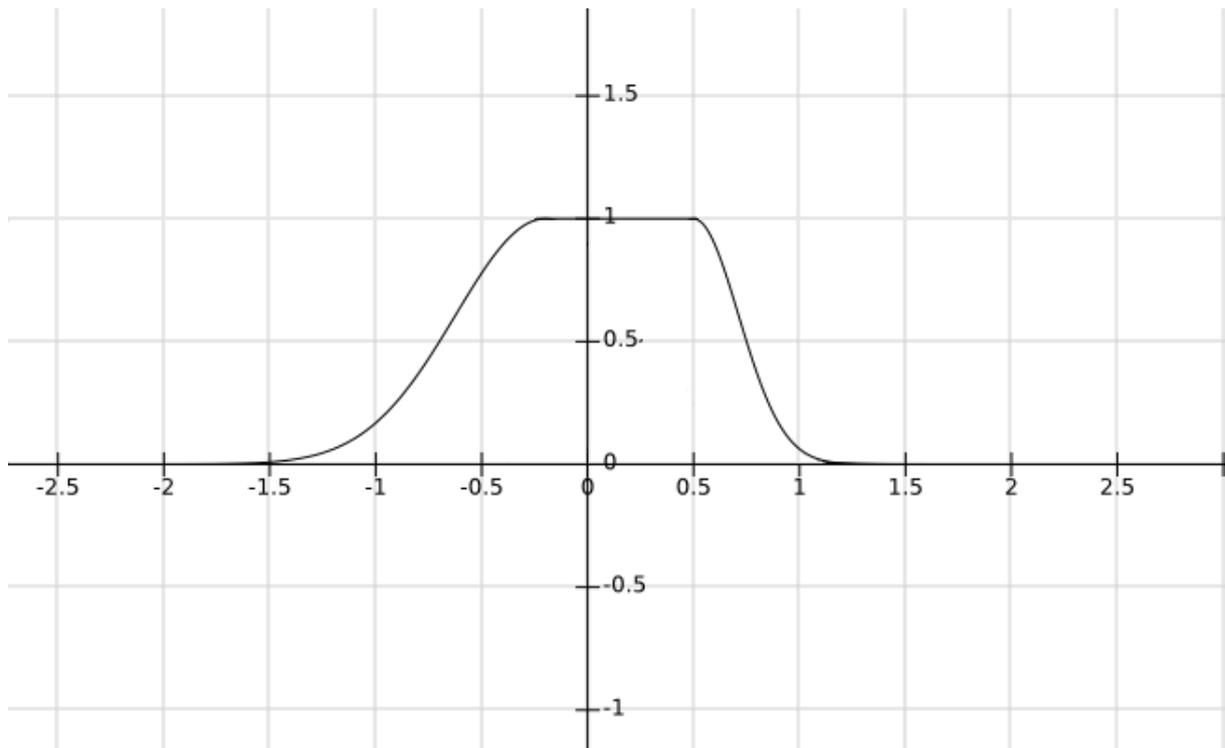


Figura 6.3: $\alpha = 0,5$ $\omega = 0,2$ $\sigma_f = 0,6$ $\sigma_n = 0,3$.

Capítulo 7

Resultados

En esta sección se muestran varias secuencias de imágenes donde se puede observar el funcionamiento del sistema en diferentes situaciones y sin modificar los parámetros del algoritmo entre ellas.

Se representan en verde los puntos de bordillo, las líneas azules largas marcan la dirección de comienzo del gusano, las líneas cortas en cada punto de bordillo marcan la dirección con la que llegó ahí el gusano, los puntos cyan son puntos de obstáculo y los puntos rojos aquellos donde el gusano establece que termina el bordillo.

En la figura 7.1 se muestra una secuencia con bordillos curvos a ambos lados, sin embargo la cámara no recoge suficiente ancho como para ver el bordillo izquierdo, desapareciendo muy pronto y además al generar el mapa de elevación la información está mal reconstruida en el bordillo izquierdo en las primeras imágenes. El bordillo derecho por el contrario está perfectamente reconstruido, consiguiendo el algoritmo delimitarlo con gran éxito en toda la secuencia.

En la figura 7.2 se muestra una secuencia en con un bordillo que se curva hacia la izquierda con la presencia de una farola que tapa la visión. El bordillo es elevado y la calidad de la reconstrucción estéreo es buena, aunque la calidad baja mucho en las zonas de bordillos no visibles. La delimitación del bordillo no es precisa en su parte final, ya que en el mapa de elevación se observa un pico que confunde al gusano. En el lado derecho hay un bordillo recto y el algoritmo funciona correctamente al tratar de delimitarlo aunque no sea su función.

En la figura 7.3 se muestra una secuencia con dos bordillos, uno recto finalizando en curva a la derecha y otro recto a la izquierda, en ambos hay un coche aparcado al final, el algoritmo de gusano no trata correctamente con obstáculos, por tanto dependiendo de la captura se comporta impredeciblemente bordeando el obstáculo o pasando entre el obstáculo y el bordillo.

En la figura 7.4 se muestra una secuencia con un bordillo recto a la izquierda y uno curvo a la derecha con la peculiaridad de que la calle a la derecha es una cuesta arriba, al calcularse las elevaciones respecto a un único plano principal la elevación de los puntos de la vía en esa calle no es cercana a cero, y se produce un desvío del gusano en el último tramo del bordillo a causa de ello.

En la figura 7.5 se muestra una secuencia con un bordillo alto a la izquierda y un bordillo bajo a la derecha con un coche aparcado, el bordillo izquierdo está perfectamente delimitado, mientras que en el lado derecho el gusano se introduce por detrás del coche al no haber un bordillo con altura, en esta secuencia se puede destacar que ese bordillo sin altura es el acceso a un vado.

En la figura 7.6 se muestra una secuencia con una serie de vehículos aparcados a la izquierda

con un bordillo de insuficiente elevación, a la derecha un bordillo que deriva en curva con una farola muy estrecha, ese bordillo se delimita perfectamente mientras que en el lado izquierdo no se encuentra bordillo excepto en la primera captura la cual trata los propios vehículos como bordillos.

En la figura 7.7 se muestra una secuencia con coches aparcados en diagonal a la izquierda y un bordillo que se curva hacia el lado a la derecha con la problemática de que hay colocados una cantidad de postes unidos por cadenas que dificultan la reconstrucción estéreo más allá de ellos en gran medida. El resultado en el lado derecho es inestable debido a la dificultad que encuentra la reconstrucción estéreo y que el propio bordillo se va aplanando con la vía en la parte más diagonal.

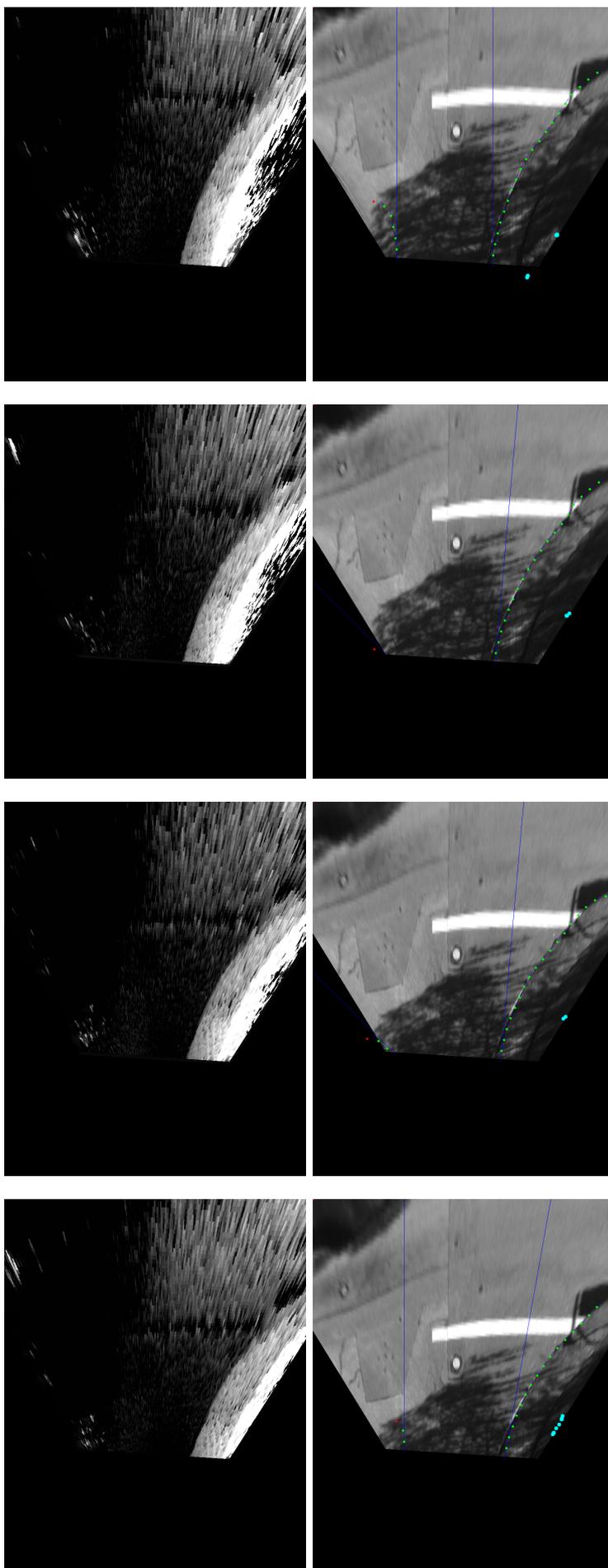


Figura 7.1: Imagen de elevaciones y resultado en vista de pájaro, secuencia 1

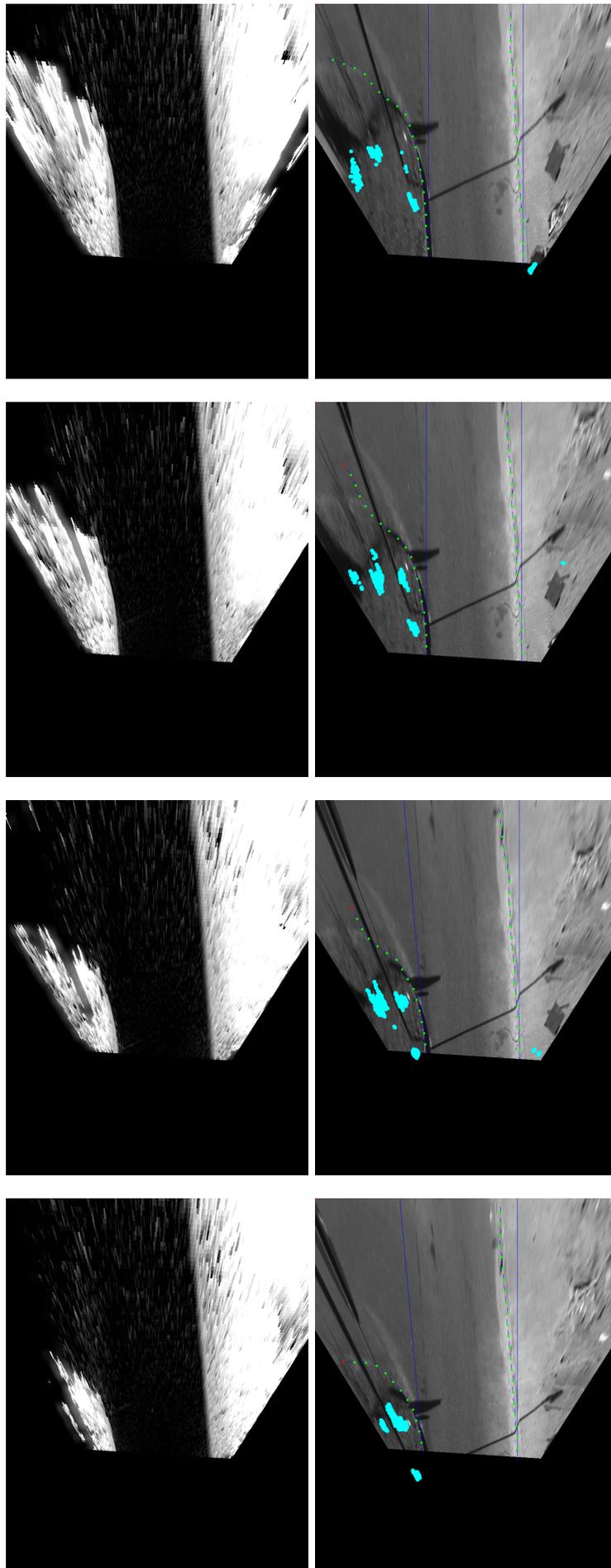


Figura 7.2: Imagen de elevaciones y resultado en vista de pájaro, secuencia 2

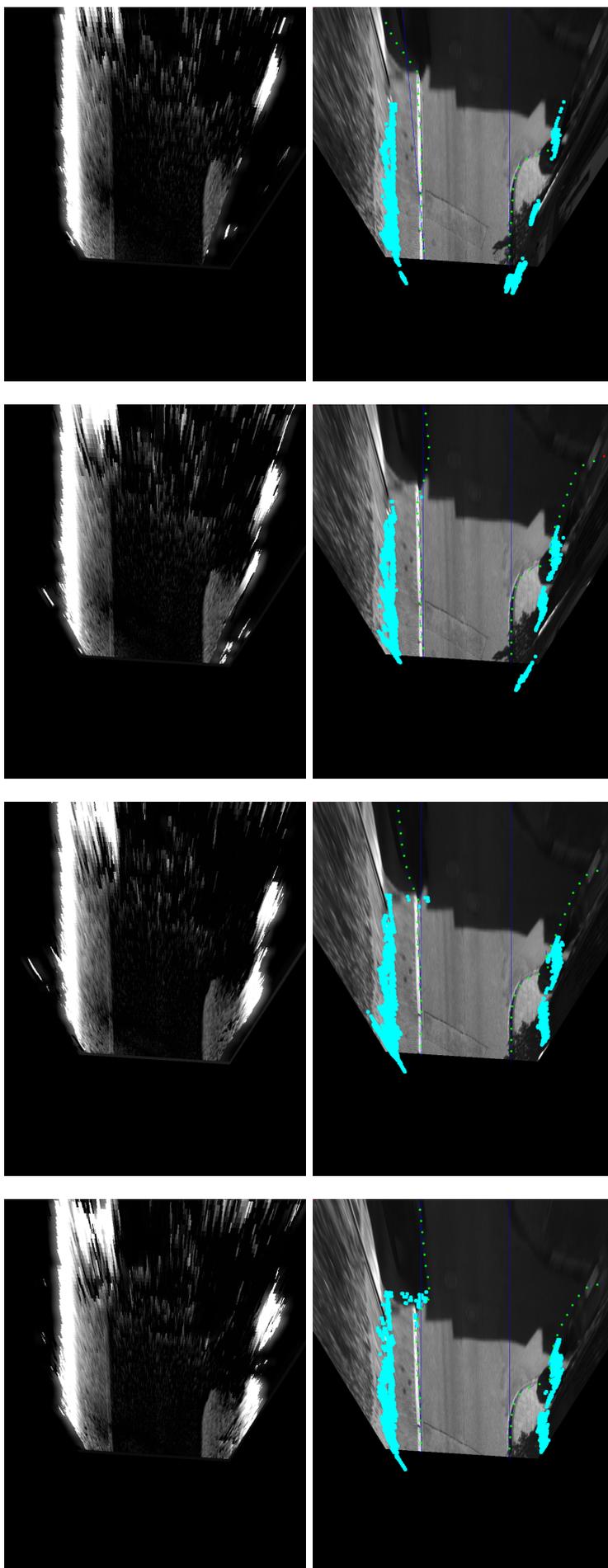


Figura 7.3: Imagen de elevaciones y resultado en vista de pájaro, secuencia 3

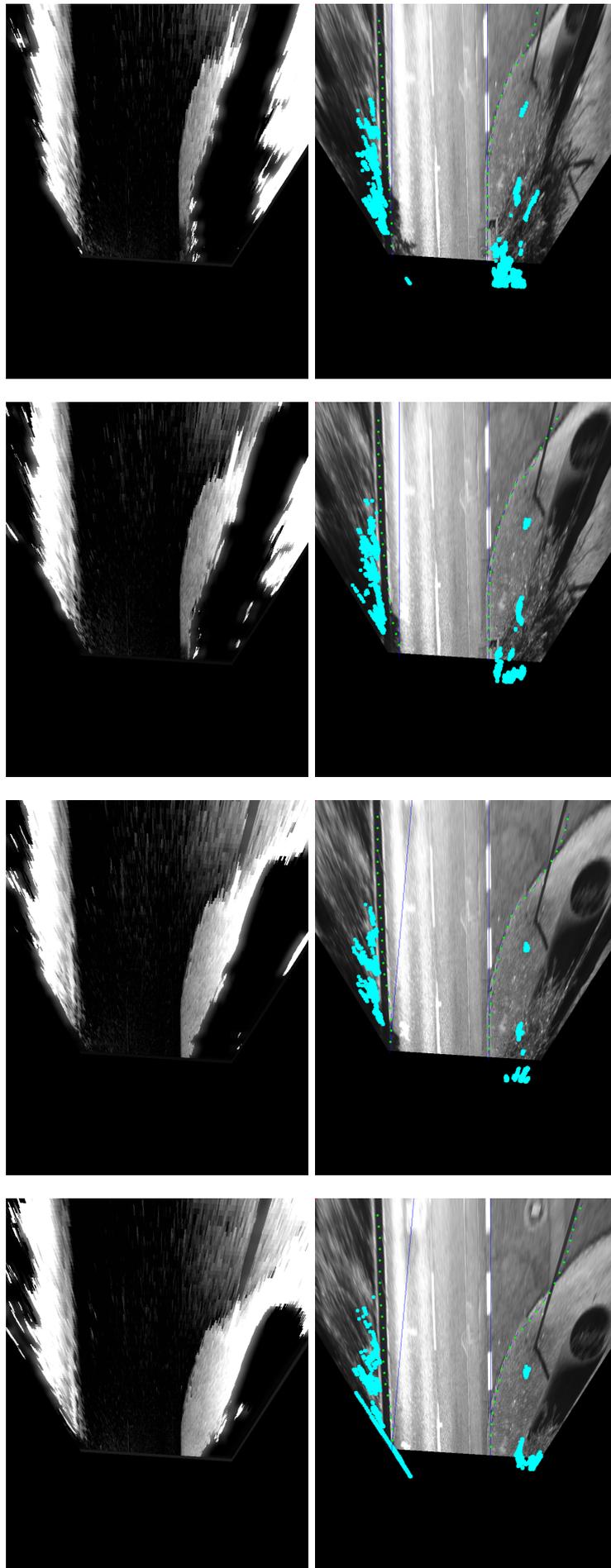


Figura 7.4: Imagen de elevaciones y resultado en vista de pájaro, secuencia 4

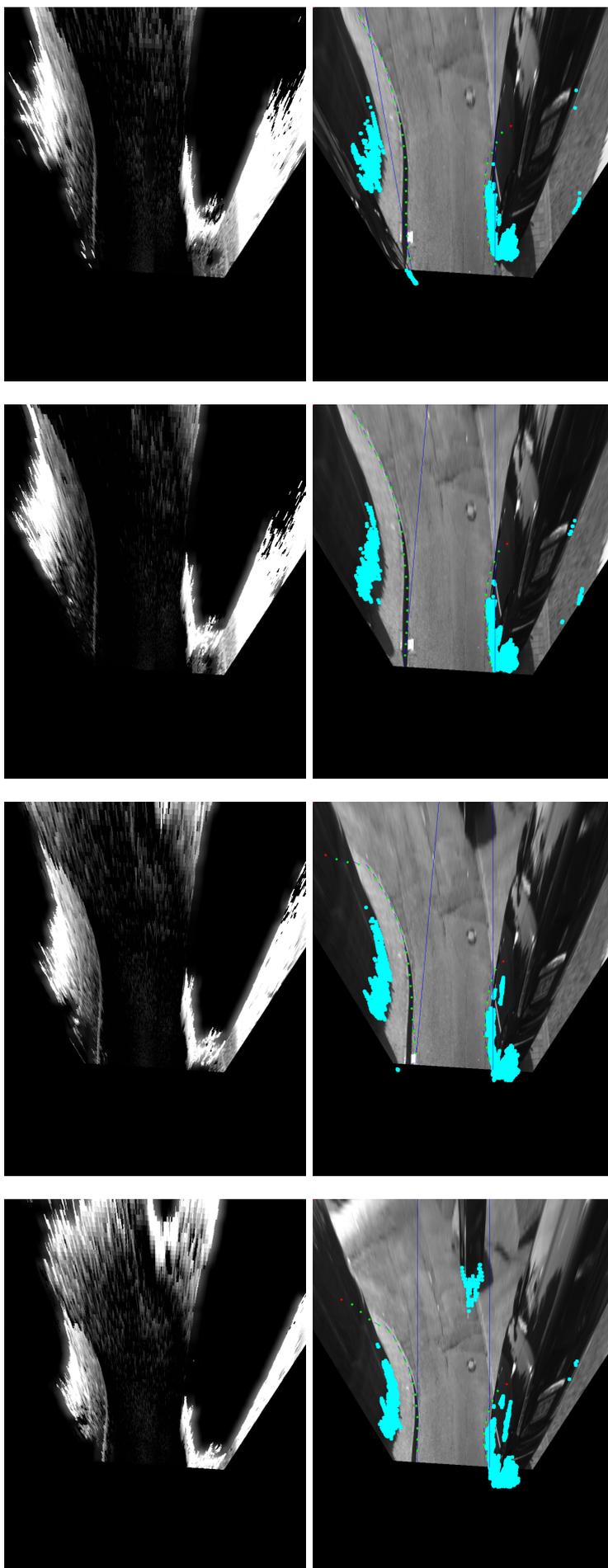


Figura 7.5: Imagen de elevaciones y resultado en vista de pájaro, secuencia 5

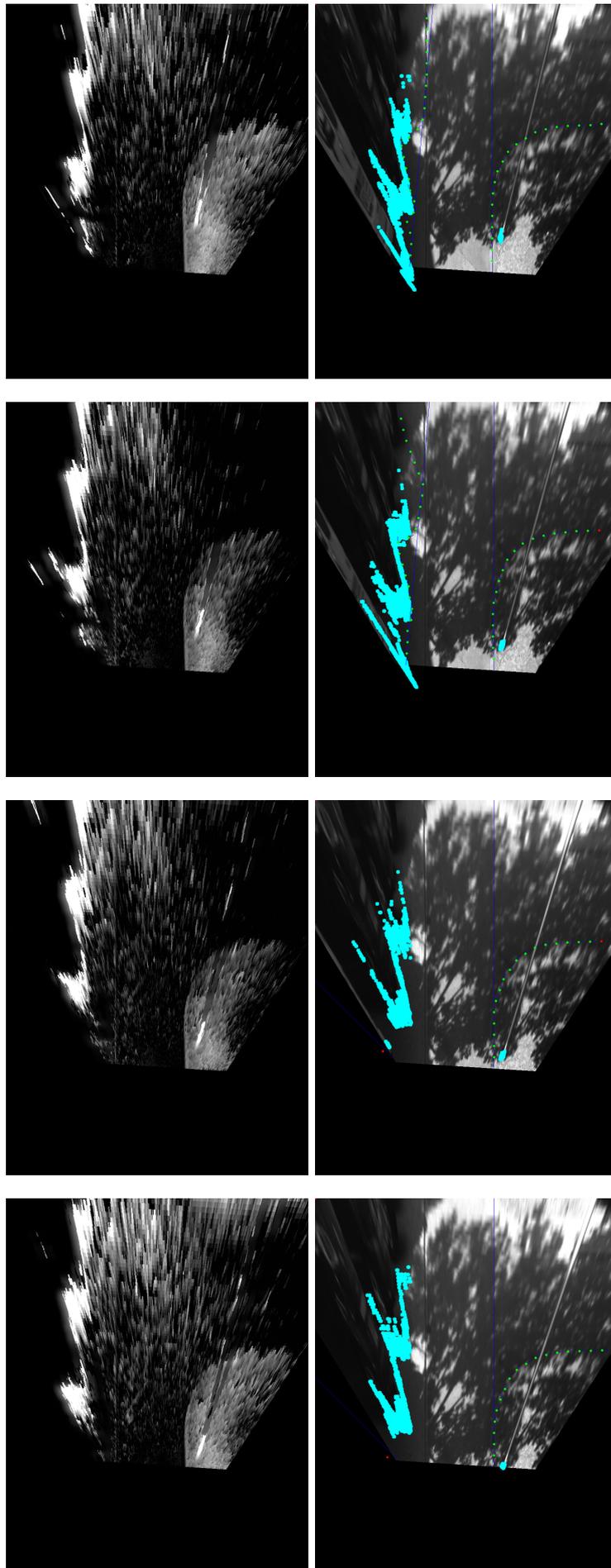


Figura 7.6: Imagen de elevaciones y resultado en vista de pájaro, secuencia 6

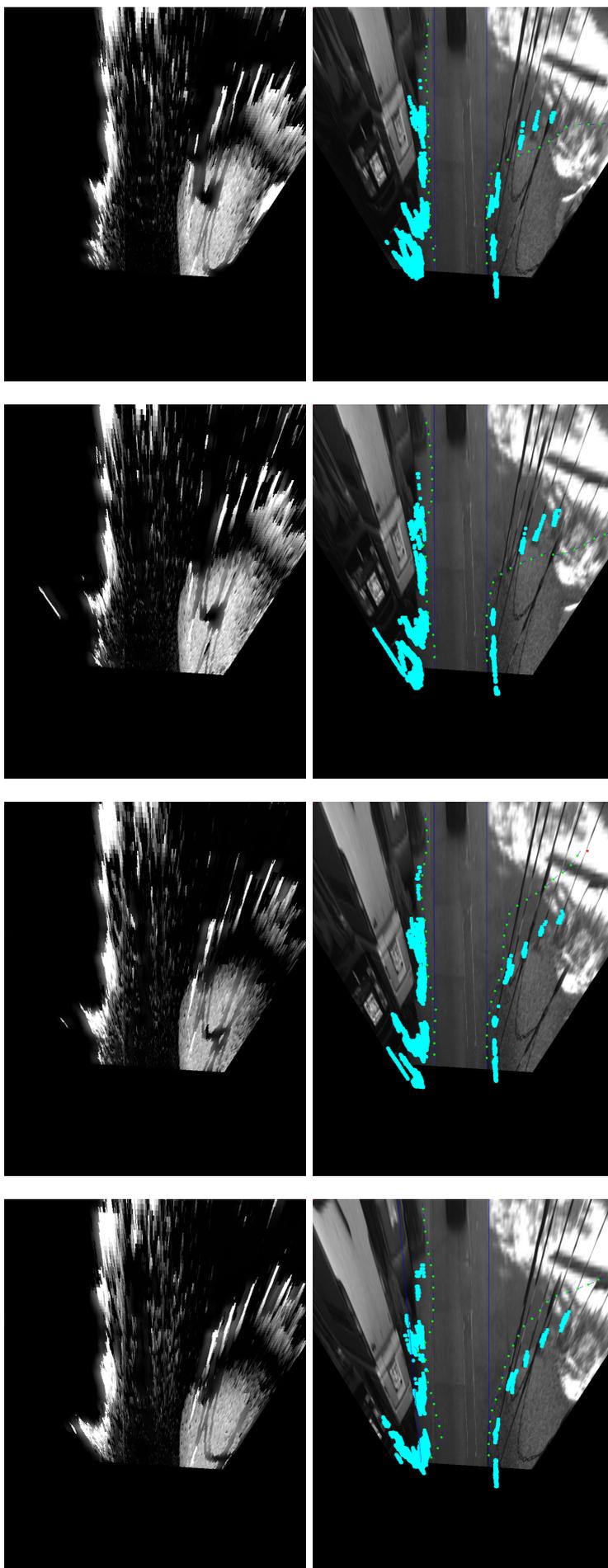


Figura 7.7: Imagen de elevaciones y resultado en vista de pájaro, secuencia 7

Capítulo 8

Conclusiones y Trabajos Futuros

A continuación se exponen las conclusiones extraíbles del trabajo realizado, y se proponen posibles futuras aportaciones que se pueden investigar o desarrollar en lo que respecta a la detección de la parte no visible de los bordillos.

8.1. Conclusiones

Las conclusiones generales que resumen el proyecto y las principales aportaciones que se han conseguido son:

- Se ha conseguido realizar a través de un método poco común una limitación de bordillos sin información tridimensional directa como su componente normal.
- Los resultados no dan una gran precisión y dependen de bordillos con una buena altura además de que la calidad de la reconstrucción estéreo no es tan buena como podría desearse y es peor en puntos lejanos donde se aprecia mejor la presencia de estos bordillos.
- Este algoritmo requiere la creación de un mapa de elevación particular para el mismo, que podría ser de utilidad en otras aplicaciones de visión para vehículos autónomos.
- El tiempo de cómputo del algoritmo en cada captura de las cámaras puede llegar a alcanzar los diez segundos, esto incluye la generación de un plano principal y el cálculo de normales, se puede optimizar en gran medida este tiempo.

8.2. Trabajos futuros

Dado el hecho de que de un trabajo con una aplicación tan concreta como ésta no se encuentran trabajos previos si bien es difícil encontrarlos, la solución propuesta está claramente sujeta a muchos cambios y mejoras, tanto en el modo de trabajar como en añadidos para mejorar la detección. Éstas son las propuestas para posibles trabajos futuros:

- Utilizar una reconstrucción estéreo de mejor calidad, ya que el resultado del algoritmo depende directamente de lo correcta que ésta sea.

- Realizar mejoras en el *hardware* y optimizar los procesos del algoritmo para reducir considerablemente el tiempo de computo y, así, hacerlo viable para su uso en sistemas de tiempo real.
- Las mediciones 3D del sistema de visión estéreo son aproximadas y en ocasiones malas, sin embargo que la información sea densa ayuda al algoritmo, empleando un sistema de medición láser, como un *lidar* se podría mejorar los resultados del algoritmo mediante una fusión de informaciones.
- El propósito de este algoritmo era originalmente la de utilizarse en bordillos visibles previamente detectados, pero para facilitar su desarrollo se ha utilizado directamente sobre la escena, se debería utilizar en bordillos visibles previamente detectados.
- Modelar el bordillo como continuación del visible previamente detectado, utilizando un modelo similar, ya que actualmente devuelve tan solo una serie de puntos que lo limitan.
- Combinar el algoritmo con una detección de obstáculos robusta y mejorarlo para que trate correctamente con la presencia de un obstáculo sobre el bordillo.

Parte III

Bibliografía

Bibliografía

- [1] "Phantom Auto will tour city", The Milwaukee Sentinel, Google News Archive, 8 December 1926. Retrieved 23 July 2013.
- [2] Reporter Rides Driverless Car", *The Press-Courier*, Google News Archive, 7 June 1960. Retrieved 21 July 2013.
- [3] M. Williams, "PROMETHEUS-the european research programme for optimising the road transport system in europe", in *Driver Information, IEE Colloquium on*, Dec 1988, pp. 1/1-1/9.
- [4] B. Ulmer, "Vita-an autonomous road vehicle (arv) for collision avoidance in traffic", in *Intelligent Vehicles '92 Symposium, Proceedings of the*, Jun 1992, pp. 36-41.
- [5] M. Bertozzi and A. Broggi, Gold: a parallel real-time stereo vision system for generic obstacle and lane detection", *Image Processing, IEEE Transactions on*, vol. 7, no. 1, Jan 1998, pp. 62-81.
- [6] M. Campbell, M. Egerstedt, J. How and R. Murray, "Autonomous driving in urban environments: approaches, lessons and challenges", *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 368, no. 1928, 2010, pp. 4649-4672.
- [7] C. Urmson, J. Anhalt, D. Bagnell, C. Baker et al., "Autonomous driving in urban environments: Boss and the Urban Challenge", *Journal of Field Robotics*, vol. 25, no. 8, 2008, pp. 425-466.
- [8] A. Geiger, P. Lenz, C. Stiller and R. Urtasun, "Vision meets robotics: The KITTI dataset", *The International Journal of Robotics Research (IJRR)*, vol. 32, no. 11, 2013, pp. 1231-1237.
- [9] G. Zhao, J. Yuan. "Curb detection and tracking using 3D-LIDAR scanner", 19th IEEE *International Conference on Image Processing (ICIP)*, 2012, pp. 437-440.
- [10] F. Oniga, S. Nedevschi and M. M. Meinecke, "Curb detection based on a multi-frame persistence map for urban driving scenarios", in *Intelligent Transportation Systems (ITSC)*, 2008 13th *International IEEE Conference on*, Oct 2008, pp. 67-72.

