

Detección de obstáculos y espacios transitables en entornos urbanos para sistemas de ayuda a la conducción basados en algoritmos de visión estéreo implementados en GPU

B. Musleh*, A. de la Escalera, J.M. Armingol

Laboratorio de Sistemas Inteligentes del Departamento de Sistemas y Automática, Universidad Carlos III de Madrid, C/Butarque, 15, 28911 Leganés, Madrid, España

Resumen

Tanto los sistemas avanzados de ayuda a la conducción (ADAS) aplicados a la mejora de la seguridad vial, como los sistemas de navegación autónoma de vehículos, demandan sensores y algoritmos cada vez más complejos, capaces de obtener e interpretar información del entorno vial. En concreto, las mayores dificultades surgen a la hora de analizar la información proveniente de los entornos urbanos, debido a la diversidad de elementos con distintas características que existen en áreas urbanas. Estos sistemas requieren, cada vez más, que la interpretación de la información se realice en tiempo real para mejorar la toma de decisiones. Por otra parte, la visión estéreo es ampliamente utilizada en sistemas de modelado, dada la gran cantidad de información que proporciona, pero al mismo tiempo, los algoritmos basados en esta técnica requieren de un elevado tiempo de cómputo que dificulta su implementación en aplicaciones de tiempo real. En este trabajo se presenta un algoritmo basado en visión estéreo para la detección tanto de obstáculos como de espacios transitables en entornos urbanos y que ha sido implementado principalmente en GPU (Unidad de Procesamiento Gráfico) para reducir el tiempo de cómputo y conseguir un funcionamiento en tiempo real. *Copyright © 2012 CEA. Publicado por Elsevier España, S.L. Todos los derechos reservados.*

Palabras Clave:

Visión por Computador, Vehículos Autónomos, Algoritmos de Detección, Sistemas de Tiempo Real.

1. Introducción

La actual demanda social de mejora de la seguridad vial es una de las grandes responsables del profundo desarrollo que han experimentado los sistemas de ayuda a la conducción. Estos sistemas normalmente realizan tareas de supervisión de la conducción hasta que se producen situaciones de peligro, generando avisos al conductor que advierten de dichas situaciones. Tradicionalmente, estos sistemas no actuaban activamente sobre elementos del vehículo tales como los frenos o la dirección, pero en los últimos tiempos esta tendencia está cambiando, llegando a ser cada vez más común encontrar soluciones básicas de este tipo implementadas comercialmente. Esto exige, si cabe aún más, un alto grado de eficacia de los algoritmos, tanto desde un punto de vista de fiabilidad, como de tiempo de respuesta, siendo un caso extremo cuando son aplicados a la navegación autónoma de vehículos (Pérez et al., 2010). El ámbito de los vehículos autónomos ha experimentado un auge a partir de

la celebración de competiciones auspiciadas por la Agencia de Investigaciones Avanzadas del Ejército Norteamericano (DARPA). Las primeras ediciones se realizaban a campo abierto y solo durante la última competición se simuló un entorno urbano, debido a la complejidad inherente a estas zonas.

Los entornos urbanos requieren del uso de sensores capaces de obtener una gran cantidad de información, para poder realizar un análisis correcto de los mismos. Por ejemplo, los *laser lidar* son sensores de uso extendido en los ADAS por la gran precisión y resolución de sus medidas, pero están limitados desde un punto de vista de la cantidad de información que suministran, ciñéndose a un plano 2D en la mayoría de los casos. En contraposición, los sistemas de visión estéreo proporcionan una gran cantidad de información en 3D, aunque con una resolución desde un punto de vista de la profundidad más limitada que los *laser lidar*. En el presente trabajo se ha empleado un sistema de percepción basado en visión estéreo que aprovecha la gran cantidad de información que ofrece.

Los entornos urbanos presentan numerosos y variados elementos en función de su tamaño, forma, color, textura, etc., lo que dificulta en gran medida el procesamiento de las imágenes del sistema estéreo. Un primer ejemplo lo encontramos en la

* Autor en correspondencia.

Correos electrónicos: bmusleh@ing.uc3m.es (B. Musleh), escalera@ing.uc3m.es (A. de la Escalera), armingol@ing.uc3m.es (J.M. Armingol)

aparición de obstáculos de gran tamaño, tales como edificios u otros vehículos, que pueden llegar a ocupar una gran parte de la imagen, ocluyendo el resto de la vía. Otra situación recurrente en entornos urbanos es la falta de textura y la aparición de patrones repetitivos en muchos de los elementos de la vía, que son captados por las cámaras y que dificultan el análisis de las imágenes por parte de los algoritmos estéreo. El cielo o la calzada carecen habitualmente de textura, siendo un caso extremo la pintura de los vehículos. Los patrones repetitivos aparecen habitualmente en los entornos urbanos, como por ejemplo en edificios de ladrillo y en las aceras. Otra fuente típica de complicaciones esta relacionada con las condiciones de iluminación, como ocurren con los brillos, la iluminación a contraluz o las sombras en la conducción diurna, así como cuando la iluminación es insuficiente, como sucede bajo condiciones meteorológicas adversas, en túneles o aparcamientos y, por supuesto, durante la conducción nocturna.

Los algoritmos han sido implementados y probados en circulación urbana mediante el vehículo experimental *IVVI*, cuyo nombre responde al acrónimo de (*Intelligent Vehicle Based on Visual Information*) (Armingol et al., 2007). Además del sistema presentado en este artículo, la plataforma posee otros sistemas de ayuda a la conducción, tales como la detección de somnolencia del conductor (Flores et al., 2011), reconocimiento de señales de tráfico, detección de peatones por infrarrojos para conducción nocturna (Olmeda et al., 2011) y un sistema de aviso de salida de carril.

En el presente trabajo se describe una nueva metodología para la detección de obstáculos y espacios transitables basada en la construcción del mapa de disparidad (Scharstein and Szeliski, 2002) y el *u-v disparity* (Hu and Uchimura, 2005). Además, se presenta un estudio de aquellas situaciones en entornos urbanos que dificultan la interpretación de los mismos. El perfil de la calzada (Labayrade et al., 2002) será utilizado para realizar una novedosa clasificación de obstáculos que permite distinguir entre aquellos que están elevados y los situados sobre el suelo (no elevados). Asimismo, se presenta una nueva formulación para la localización de obstáculos sobre la calzada que aumenta considerablemente la resolución de la misma. Respecto a la implementación del algoritmo, una gran parte del mismo ha sido realizada en CUDA (*Compute Unified Device Architecture*) para poder procesar en GPU y de esta manera reducir sensiblemente el tiempo de cómputo hasta alcanzar una respuesta en tiempo real. En aplicaciones de tipo ADAS se entiende por tiempo real el tiempo de procesado de la información que permita emitir avisos al conductor con tiempo suficiente para que éste detenga el vehículo ante una situación de peligro.

1.1. Estructura del Documento

La sección segunda presenta el estado del arte de la detección de obstáculos y de la determinación del espacio libre en su aplicación a los sistemas de ayuda a la conducción. A continuación, en la tercera se describe el algoritmo, donde se explicará cómo se detectan los obstáculos y las zonas transitables, la obtención y usos del perfil de la calzada y el estudio de casos complejos en entornos urbanos. El apartado cuarto se dedica a detallar la implementación del algoritmo, prestando una mayor

atención a la parte correspondiente a la GPU. Se presentan los resultados experimentales obtenidos en las pruebas realizadas en la sección quinta y, finalmente, en el punto sexto se exponen las conclusiones.

2. Estado del Arte

Actualmente se utilizan distintas tecnologías para la detección de obstáculos y la determinación del espacio libre aplicadas a los sistemas de ayuda a la conducción. En este apartado se mostrarán los principales trabajos realizados en este campo y a lo largo del artículo se referenciarán aquellos que están más relacionados con las técnicas utilizadas. Dentro de estas tecnologías cabe diferenciar entre dos grandes grupos: en primer lugar están los sistemas basados en tecnologías láser o radar, que se caracterizan por proveer una cantidad bastante reducida de información, además de ser sensores invasivos. Por otro lado encontramos los basados en visión por computador, que principalmente están implementados mediante sistemas monoculares o sistemas estéreo y en los que, al contrario que los anteriores, la cantidad de información adquirida es muy superior y son no invasivos. Por su parte, los sistemas monoculares presentan la desventaja frente a los estéreo de no disponer información 3D.

Se han conseguido buenos resultados en sistemas de detección de obstáculos basados en *laser lidar* (Broggi et al., 2008) (Urmson et al., 2008), principalmente debido a la alta precisión de sus medidas, aunque presentan dificultades a la hora de realizar clasificaciones de los obstáculos detectados debido a la escasa información que adquieren (Premebida et al., 2009) (Petrovskaya and Thrun, 2009). Los *laser lidar* también se pueden utilizar para determinar las zonas libres de obstáculos mediante algoritmos de celdas de ocupación (Thrun, 2003).

Los sistemas monoculares se utilizan habitualmente en la detección y clasificación de diferentes tipos de obstáculos, principalmente peatones (Enzweiler and Gavrilu, 2009) y vehículos (Sun et al., 2006). Existe una gran variedad de algoritmos aplicados a estas tareas y que también dependen de las condiciones de iluminación. Por ejemplo, para la detección de peatones en conducción nocturna en los últimos años se ha extendido el uso de cámaras infrarrojas de espectro lejano (Bertozzi et al., 2003), aunque también se pueden utilizar métodos de iluminación, preferiblemente en el infrarrojo cercano (Ling et al., 2007) a fin de evitar ocasionar deslumbramientos al resto de usuarios de la vía. Por último, existen algoritmos aplicados a la detección del espacio libre empleando este tipo de sensores (Taylor et al., 2004). Los sistemas estéreo aportan, respecto de los monoculares, información 3D, pero los algoritmos deben procesar una mayor cantidad de información. Son muchos los trabajos que utilizan visión estereoscópica en el ámbito de la detección de obstáculos (Xu et al., 2009) (Cabani et al., 2006) (Lemond and Devy, 2004), siendo también numerosos los que hacen un uso específico del *u-v disparity* (Hu et al., 2005) (Labayrade and Aubert, 2003a) (Hautière et al., 2006). La detección de las zonas libres (Guo et al., 2009) (Soquet et al., 2007) supone un campo de investigación muy intenso utilizando esta tecnología, que usa normalmente como herramienta el perfil de la calzada (Zhao et al., 2009) (Labayrade and Aubert, 2003b).

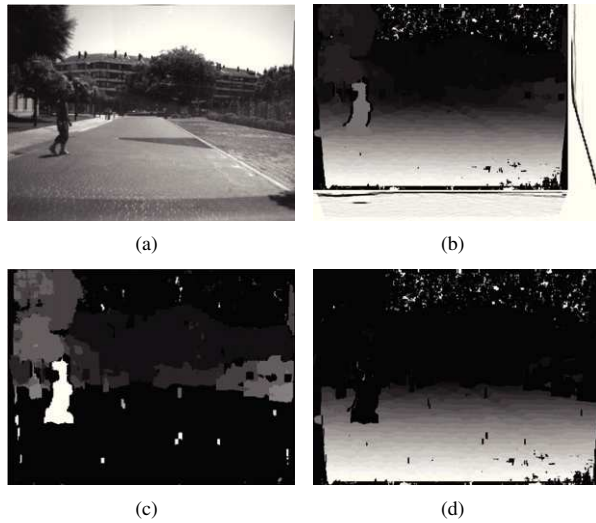


Figura 1: Ejemplo de análisis de un entorno urbano para la detección de obstáculos y espacio libre. (a) Visible izquierda. (b) Mapa denso de disparidad. (Derecha) v -disparity. (Debajo) u -disparity. (c) Mapa de obstáculos. (d) Mapa libre.

3. Descripción del Algoritmo

Para calcular la profundidad (Z) a la que se encuentra un punto $P = (X, Y, Z)$ en coordenadas del mundo es necesario utilizar dos cámaras que posean ciertas características que conformen lo que se denomina un par estéreo. Disponiendo de un par de cámaras iguales y paralelas se puede utilizar la ecuación estéreo (1) para calcular la profundidad (Z) del punto P , donde la proyección del punto P sobre el plano de la imagen izquierda es (u_L, v_L) y sobre el plano de la imagen derecha es (u_R, v_R) , f es la distancia focal medida en píxeles, B es la *baseline* entre cámaras y d es la disparidad.

$$Z = \frac{f \cdot B}{u_L - u_R} = \frac{f \cdot B}{d} \quad (1)$$

Construir un sistema estéreo que cumpla las condiciones canónicas es muy difícil en la práctica y por tanto se debe aplicar una etapa de rectificación (Fusiello et al., 2000) sobre las imágenes para que se cumpla (1).

El mapa de disparidad (Fig. 1(b)) es construido a partir de las imágenes obtenidas de las cámaras (Fig. 1(a)), donde la disparidad (d) aparece representada por niveles de gris para cada punto de la imagen, donde los más cercanos poseen un nivel de gris mayor que los más alejados. Una vez que se ha construido el mapa denso de disparidad se puede obtener el u - v disparity (Fig. 1(b) debajo y derecha). El v -disparity corresponde al histograma de la disparidad para cada una de las filas del mapa de disparidad, mientras que el u -disparity corresponde también al histograma, pero en este caso de cada columna. El u - v disparity es una fuente de información muy completa puesto que, en primer lugar los obstáculos que hay delante del vehículo aparecen como líneas proporcionales a las dimensiones de los mismos y en su valor correspondiente de disparidad (Broggi et al., 2005). Además de la información acerca de los obstáculos, también es

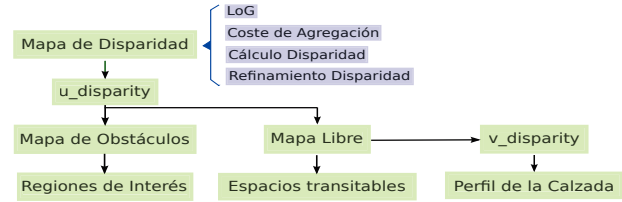


Figura 2: Esquema del diagrama de flujo del algoritmo.

posible obtener información de la calzada, la cual aparece en el v -disparity como una línea oblicua (Labayrade et al., 2002). Esta línea se puede expresar con la ecuación de una recta (2) si se realiza la asunción de suelo plano delante del vehículo, donde m es la pendiente y b es la ordenada en el origen, que corresponde al valor teórico del horizonte en la imagen. Una vez determinados los valores de la expresión se puede calcular el ángulo de cabeceo (α) entre el sistema estéreo y la calzada, cuyo valor viene dado por (3), siendo C_v la coordenada vertical del centro óptico.

$$v = m \cdot d + b \quad (2)$$

$$\alpha = \arctan \frac{b - C_v}{f} \quad (3)$$

El mapa de disparidad y el u - v disparity se emplearán para determinar los obstáculos y el espacio libre delante del vehículo; una aplicación extendida consiste en la utilización de la transformada de Hough para detectar los obstáculos como líneas (Hu and Uchimura, 2005), pero resulta costoso desde un punto de vista de tiempo de cómputo. El objetivo de nuestra detección de obstáculos es la determinación de las regiones de interés que rodean a los mismos y podrán ser clasificados como elevados o no elevados por medio de la utilización del perfil de la calzada. Esta clasificación tiene, entre otras posibles aplicaciones, la de llegar a determinar si el vehículo puede transitar por debajo de los obstáculos elevados, como son las entradas a los túneles, pórticos o semáforos. Además, se presenta un método que combina la ecuación estéreo (1) con el perfil de la calzada (2), lo que permite calcular la profundidad (Z) a la que se encuentran los obstáculos con una resolución muy superior a la obtenida usando únicamente los valores de disparidad en (1).

Como se detalló anteriormente, en los entornos urbanos existe un elevado número de elementos que presentan grandes diferencias en función de su tamaño, forma, localización, color, textura, etc, lo cual afecta al mapa de disparidad y por tanto, al u - v disparity (Lee et al., 2008). En este artículo se presentan los resultados de la interpretación del mapa de disparidad y del u - v disparity para situaciones comunes en entornos urbanos, donde existen dificultades para interpretar la información de manera correcta. Además se expone una metodología que permite reducir el número de ocasiones en que el perfil de la calzada es detectado de manera incorrecta, que es el problema más común en estas situaciones complejas.

La Fig. 2 presenta un diagrama de flujo del algoritmo donde se pueden apreciar las distintas etapas del mismo y las secuencias de ejecución.

3.1. Detección de Obstáculos y de Zonas Transitables

La información contenida en el mapa denso de disparidad puede dividirse en dos grupos: obstáculos y espacio libre, obteniéndose como resultado otros dos nuevos mapas densos de disparidad. El primero de ellos, el mapa de obstáculos, contiene únicamente los píxeles del mapa de disparidad correspondientes a obstáculos (Fig. 1(c)). Mientras que el segundo, el mapa libre, es el opuesto al anterior donde solamente aparece el espacio libre existente delante del vehículo (Fig. 1(d)). Como se explicará más adelante, el mapa de obstáculos constituye la base para la determinación de las regiones de interés que engloban a los distintos obstáculos, mientras que el mapa libre se empleará para la obtención del perfil de la calzada.

3.1.1. Construcción del Mapa de Obstáculos y del Mapa Libre

Para la construcción tanto del mapa de obstáculos como del mapa libre, se parte del mapa denso de disparidad y del u - v disparity. Como se indicó anteriormente, los obstáculos aparecen en el u -disparity como líneas, siendo esta información la base para generar tanto el mapa de obstáculos como el mapa libre.

Así pues, la construcción del mapa de obstáculos se realiza en dos fases: en primer lugar, se umbraliza el u -disparity a fin de detectar todos aquellos obstáculos cuya altura sea mayor a un umbral medido en píxeles y cuyo valor debe elegirse con cuidado, pues tendrá que ser lo suficientemente pequeño para detectar cualquier obstáculo que pudiera impedir el movimiento del vehículo, pero suficientemente grande para evitar detectar como obstáculos aquellas variaciones pronunciadas de la calzada que no impidan el avance del vehículo, como por ejemplo los badenes. Una vez realizada esta primera fase, el siguiente paso consiste en eliminar del mapa denso de disparidad todos aquellos píxeles que no pertenezcan a los obstáculos detectados en el u -disparity umbralizado, obteniéndose como resultado el mapa de obstáculos (Fig. 1(c)). La construcción del mapa libre (Fig. 1(d)) reutiliza el resultado de la primera etapa de la generación del mapa de obstáculos y en la segunda etapa, al contrario que para el mapa de obstáculos, se conservan los píxeles que no pertenecen a aquéllos, obteniendo así el mapa libre, cuyo resultado corresponderá normalmente a la calzada delante del vehículo, aunque teóricamente es todo espacio libre delante del mismo.

3.1.2. Determinación de las Regiones de Interés

Como se ha explicado anteriormente, para la determinación de las regiones de interés sobre la imagen visible se utiliza el mapa de obstáculos como punto de partida, sobre el que se aplicará un análisis de *blobs*. En la primera fase se fijará la región de estudio delante del vehículo sobre la que se van a buscar los obstáculos y que viene fijada por un valor de disparidad (d), de manera que aquellos obstáculos cuyo valor sea menor al de disparidad, no serán englobados en ninguna región de interés. La manera práctica de lograr esto es mediante la umbralización del mapa de obstáculos, utilizando como umbral el valor de disparidad que determina la región de estudio, resultando así un mapa de obstáculos umbralizado en el que solo aparecerán aquellos que estén dentro de la región de estudio.

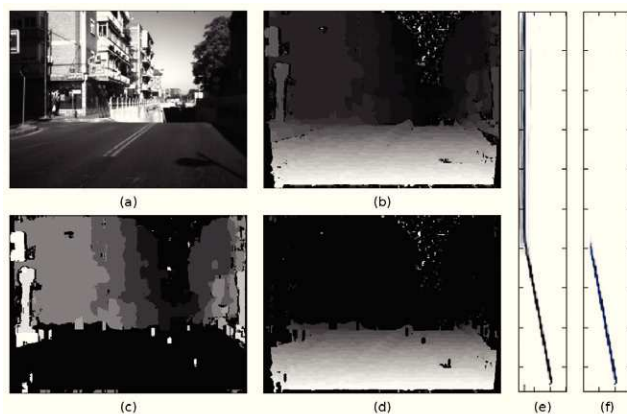


Figura 3: Comparación en la obtención del perfil de la calzada a partir del mapa de disparidad y su obtención a partir del mapa libre. (a) Imagen visible izquierda. (b) Mapa de disparidad. (c) Mapa de obstáculos. (d) Mapa libre. (e) v disparity a partir del mapa de disparidad. (f) v disparity a partir del mapa libre.

Al umbralizar, el nivel de gris de todos los píxeles pasa a ser uno para los obstáculos y cero para el resto, razón por la cual dos o más obstáculos que estén próximos o que se ocluyan parcialmente entre sí no podrán ser diferenciados por su nivel de gris o, lo que es lo mismo, por su nivel de disparidad. Los grupos de obstáculos en los que se dé esta circunstancia serán englobados en una única región de interés por el análisis de *blobs*, lo que dificultaría una posterior etapa de clasificación. A efectos de englobar cada obstáculo en una única región de interés, se lleva a cabo una detección de bordes sobre el mapa de obstáculos antes de la umbralización y se sustrae el resultado del mapa de obstáculos umbralizado, de forma que desaparece la posible agrupación de obstáculos. Es importante mencionar que este procedimiento puede ocasionar la división de los obstáculos que estén muy cerca del vehículo, debido a que en esta zona un obstáculo puede tener varios valores de disparidad.

3.2. Obtención y usos del Perfil de la Calzada

Tal como se ha visto anteriormente, el perfil de la calzada aparece como una línea oblicua en el v -disparity. Un método para obtener el perfil de la calzada es la aplicación de la transformada de *Hough* para rectas. Este método obtiene como resultado la recta más votada o, lo que es lo mismo la recta, que contiene una mayor cantidad de puntos existentes en el v -disparity. Así pues, surgen problemas cuando el perfil de la calzada no es la recta con más puntos (Fig. 3(e)), tal como sucede por ejemplo cuando uno o varios obstáculos de gran tamaño aparecen delante del vehículo, lo cual es muy frecuente en entornos urbanos. Es por ello que el método presentado para la obtención del perfil de la calzada emplea un v -disparity diferente (Fig. 3(f)), caracterizado por el hecho de que los obstáculos se han eliminado del mismo. Se puede utilizar el mapa libre (Fig. 3(d)) para obtener este nuevo v -disparity, que como se ha explicado, es un mapa denso de disparidad donde todos los píxeles pertenecientes a los obstáculos se han eliminado. De esta forma, se reduce en gran medida el número de ocasiones en las que el perfil de la calzada no es la más votada en el v -disparity.

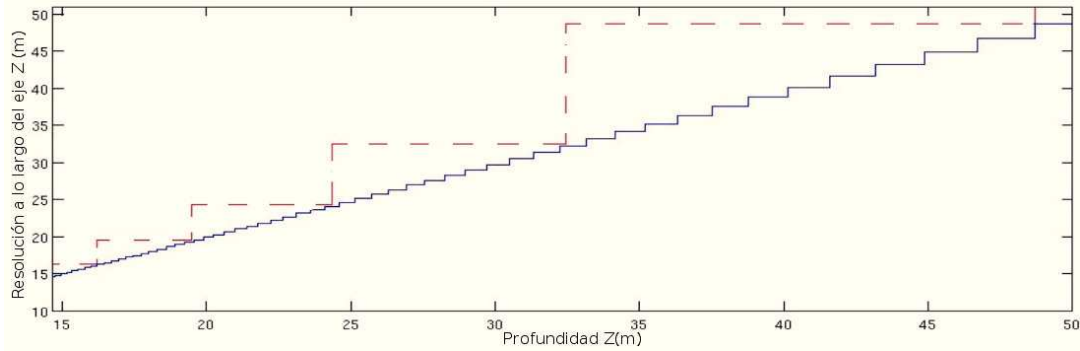


Figura 4: Comparación de la resolución usando los valores de disparidad (línea roja discontinua) frente a emplear el perfil de la calzada (línea azul)

3.2.1. Clasificación de las Regiones de Interés

Todas las regiones que son detectadas siguiendo el algoritmo anteriormente explicado, van a ser clasificadas en dos grupos: obstáculos elevados y no elevados, clasificación que puede ser de gran ayuda ya que permite descartar regiones de interés a efectos de posteriores clasificaciones más concretas como, por ejemplo, una posterior clasificación de peatones como la presentada en (Dalal and Triggs, 2005), donde aquellos que aparecen delante del vehículo solo pertenecen al grupo de obstáculos no elevados, de forma que únicamente las regiones de interés preclasificadas como no elevadas pasarán a la siguiente etapa de clasificación (este mismo procedimiento es aplicable a los algoritmos de clasificación de vehículos). Por otro lado, las regiones preclasificadas como elevadas pueden ser utilizadas por algoritmos de clasificación de señales de tráfico. Este método de elección de las regiones de interés para cada uno de los sistemas de clasificación puede reducir considerablemente el coste computacional, pues disminuye en gran medida la cantidad de información a procesar por las mismas. Otra posible aplicación de este método de clasificación es llegar a determinar cuándo un vehículo podría llegar a circular por debajo de obstáculos elevados, tal y como sucede habitualmente al aproximarse a la entrada de un túnel o de pódicos.

Para poder realizar la clasificación entre obstáculos elevados o no elevados, es necesario estimar un valor teórico de la disparidad para cada uno de ellos en el caso hipotético de que todos se encontraran sobre el suelo o calzada. Para estimar este valor se va a hacer uso de (2), utilizando el valor de la coordenada vertical v inferior de la región de interés que engloba a cada uno de los obstáculos. Para aquellos cuyo valor de disparidad teórica coincida con la real, se podrá concluir que no están elevados, mientras que si el valor de disparidad teórico es inferior al real, los obstáculos serán elevados. Para aquellos obstáculos elevados que se encuentran en la dirección de circulación del vehículo, se deberá determinar si éste puede pasar por debajo. La estimación del espacio entre los obstáculos y el suelo (h) se puede realizar mediante (4), que expresa este espacio como un valor proporcional a la distancia al obstáculo y la diferencia entre la coordenada vertical de éste v_{obs} y la coordenada vertical v_{suelo} del punto de intersección entre la proyección perpendicular del mismo sobre el suelo. Para obtener v_{suelo} se puede

utilizar el perfil de la calzada (2) con el valor de disparidad (d) predominante en el obstáculo.

$$h = \frac{B}{d}(v_{suelo} - v_{obs}) = \frac{B}{d}((m \cdot d + b) - v_{obs}) \quad (4)$$

3.2.2. Localización de Obstáculos

Utilizando el perfil de la calzada es posible localizar con una mayor resolución todos aquellos obstáculos que han sido clasificados como no elevados. Es decir, para los obstáculos elevados únicamente se puede calcular la profundidad (Z) a la que se encuentran como una función de la disparidad d mediante (5). Sin embargo, para los no elevados se cumple la relación dada por el perfil de la calzada (2) entre la coordenada vertical de la imagen v y la disparidad d . La localización en coordenadas del mundo (X, Z) de los obstáculos no elevados que se encuentran delante del vehículo puede obtenerse como una función de las coordenadas de la imagen izquierda (u, v), para lo cual es necesario combinar las ecuaciones (5) y el perfil de la calzada (2), resultando (6), donde C_u corresponde a la coordenada horizontal del centro óptico. Como se puede observar en ambas expresiones ((5) y (6)) se multiplica por $\cos \alpha$, esto es así debido a que se desea calcular la localización de los obstáculos respecto del vehículo y no respecto del sistema estéreo. Si recordamos, α corresponde al ángulo de cabeceo, que es el ángulo que forman el sistema estéreo y la calzada, y puede calcularse para cada *frame* mediante (3).

$$Z = \frac{f \cdot B}{d} \cdot \cos \alpha \quad X = Z \cdot \frac{u - C_u}{f} \cdot \cos \alpha \quad (5)$$

$$Z = \frac{m \cdot f \cdot B}{v - b} \cdot \cos \alpha \quad X = \frac{m \cdot B \cdot (u - C_u)}{v - b} \cdot \cos \alpha \quad (6)$$

La Fig. 4 muestra una comparación de la resolución de la profundidad (Z) entre los dos métodos de localización. La línea roja discontinua refleja el resultado de utilizar solo los valores de disparidad (5), mientras que la línea continua azul es producto de emplear el perfil de la calzada (6). Tal y como muestra el gráfico, la resolución resultante de emplear el perfil de la calzada mejora considerablemente con respecto a utilizar exclusivamente los valores de disparidad. Es importante resaltar que la resolución de los obstáculos no es lineal, esto es, la resolución empeora cuando los obstáculos se alejan del vehículo.

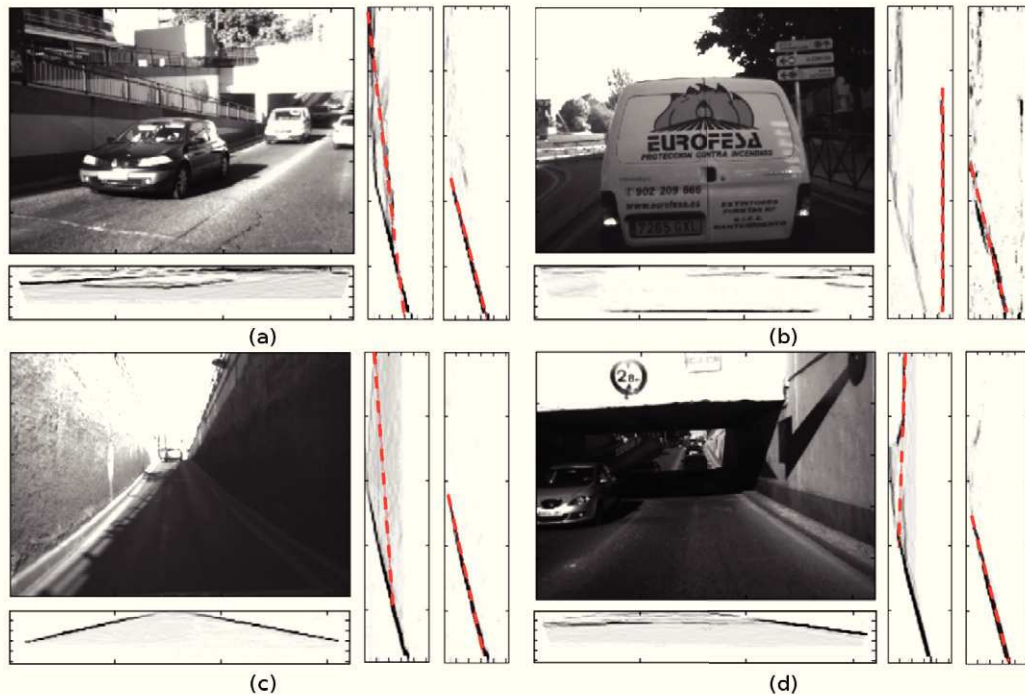


Figura 5: Ejemplos del $u-v$ disparity en diferentes casos de estudio en entornos urbanos: (a) Un vehículo y un edificio. (b) Atasco. (c) Circulación entre muros. (d) Ejemplo de obstáculo elevado: entrada a un túnel.

3.3. Casos de Estudio en Entornos Urbanos

En esta sección se tratan diferentes situaciones comunes en entornos urbanos, cuyo interés radica en el hecho de que presentan características que dificultan el análisis de la información obtenida por el sistema de percepción.

3.3.1. Obstáculos de Gran Tamaño Delante del Vehículo

La primera situación y una de las más frecuentes se da cuando un obstáculo de gran tamaño aparece delante del vehículo, como en el caso de otro vehículo o un edificio. La Fig. 5 muestra dos casos diferentes donde aparece un vehículo delante. La (Fig. 5(b)) es una imagen típica de atasco en la vía, mientras que en la (Fig.5(a)) aparece un vehículo aproximándose y una pared detrás. El problema surge en principio porque la línea del obstáculo es la más importante en el v -disparity, debido a que el obstáculo abarca una región importante de la imagen. En ambos casos, el u -disparity se presenta debajo de las imágenes visibles, donde es posible distinguir con claridad los obstáculos como líneas. Asimismo, se muestran dos v -disparity diferentes para cada caso: el primero (izquierda) se genera usando el mapa de disparidad, siendo erróneo el perfil de la calzada obtenido (línea roja). Sin embargo, en el segundo caso (derecha) el v -disparity se ha generado a partir del mapa libre, en el que no hay obstáculos, por lo que el perfil de la calzada es ahora correcto. En estos casos la construcción del mapa de disparidad también puede verse afectada, debido a que los vehículos presentan casi una ausencia total de textura.

El siguiente ejemplo refleja un caso especial: dos grandes muros situados a ambos lados del vehículo, como muestra la

Fig. 5(c). Ambos muros aparecen claramente en el u -disparity como dos largas líneas oblicuas y, por tanto, son detectadas fácilmente como obstáculos. El problema radica en que los dos muros aparecen también en el v -disparity, por lo que el perfil de la calzada obtenido en el v -disparity generado a partir del mapa de disparidad es erróneo. El mapa libre y el nuevo v -disparity se construyen con objeto de solventar este problema y así poder hallar el perfil de la calzada correctamente. El hecho de pasar por debajo de obstáculos elevados tales como semáforos o la entrada a un túnel (Fig. 5(d)) es una acción típica en la circulación de vehículos, pero no es posible distinguir si el obstáculo interrumpe el movimiento del vehículo a partir únicamente del estudio del u -disparity. Para ello es necesario utilizar la información del perfil de la calzada, como se explicó en el apartado de clasificación.

3.3.2. Túneles

En los túneles surgen diversos problemas. La construcción del mapa de disparidad es compleja debido a la débil iluminación existente, así como por el hecho de que hay muchas áreas sin textura tales como muros, calzada y techo (Fig. 6(a)). Generar el mapa de disparidad (Fig.6(b)) en estas condiciones puede mejorarse mediante el incremento del tamaño de la ventana de agregación en la etapa de agregación de coste (Wang et al., 2006). Una vez que el mapa de disparidad es suficientemente bueno, el siguiente problema surge en el $u-v$ disparity debido a que el techo del túnel también aparece en el v -disparity (izquierda) como una línea larga oblicua y podría detectarse como si fuera el perfil de la calzada. Sería posible obtener correctamen-

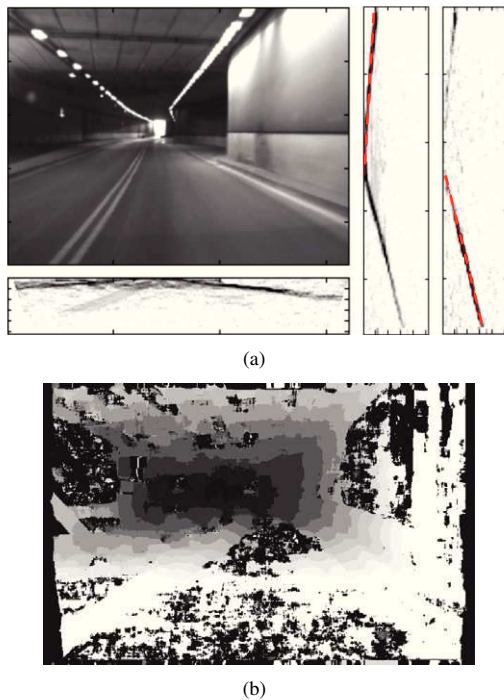


Figura 6: (a) Análisis del $u-v$ disparity en el interior de un túnel. (b) Ejemplo del mapa de disparidad en el interior de un túnel.

te el perfil de la calzada si el techo se elimina del v -disparity. No es sencillo detectar el techo como obstáculo en el caso de los túneles porque la probabilidad de detectar entonces la calzada como obstáculo también aumenta. Por tanto, una solución posible consiste en intentar disminuir su contribución tanto como sea posible en el nuevo v -disparity (Fig. 6(a) derecha).

4. Implementación del Algoritmo

Los algoritmos de visión estereó normalmete son computacionalmente muy exigentes, un ejemplo lo encontramos en la construcción del mapa de disparidad. Por esta razón se ha implementado buena parte del algoritmo utilizando NVIDIA CUDA (Nvidia, 2008) para procesar el algoritmo en la tarjeta gráfica o GPU, lo que reduce el tiempo de cómputo de manera muy significativa. Para que la implementación en CUDA sea eficaz, es necesario que los algoritmos sean susceptibles de un alto grado de paralelización. Así pues, las partes del algoritmo con un bajo grado de paralelización se han implementado en la CPU permitiéndose así una concurrencia entre la GPU y la CPU. Cada uno de los hilos de un algoritmo indicado para que su implementación en GPU sea eficaz, ha de tener las siguientes características (Kim et al., 2009):

1. Poseer un acceso exclusivo a la memoria de almacenamiento.
2. No usar resultados de otros hilos.
3. No existan condiciones de flujo en función de los datos.

Estas características son muy restrictivas, pero el sistema NVIDIA CUDA reduce dichas restricciones (NVIDIA, 2009), facilitando así el trabajo.

De esta manera, la GPU es la encargada de construir el mapa de obstáculos y el mapa libre, generando el correspondiente mapa denso de disparidad y el $u-v$ disparity, ya que todas estas partes del algoritmo tienen un alto grado de paralelización. Por su parte, la CPU se encarga de obtener el perfil de la calzada y realiza el análisis de blobs, que son operaciones con un menor grado de paralelización.

4.1. Implementación en la GPU mediante CUDA

Como se ha visto anteriormente, el mapa denso de disparidad representa la disparidad (d) de cada punto de la imagen, y como muestra (1) para determinar la disparidad ($d = u_L - u_R$) es necesario determinar la correspondencia entre puntos de la imagen izquierda y derecha respectivamente, lo que se denomina problema de correspondencia estereó. En el caso de que ambas imágenes estén correctamente rectificadas, estaremos ante un problema 1D. El algoritmo sigue la taxonomía presentada por (Scharstein and Szeliski, 2002) para resolver el problema de correspondencia estereó, donde se propone la realización de cuatro etapas: la primera es el cálculo de la función de coste (Brown et al., 2003), donde se ha utilizado el método Diferencias Cuadradas (SD) por ser más rápido de implementar en CUDA. El método SD asume una ganancia igual en ambas cámaras, razón por la cual se realiza un preprocesado de las imágenes del par estereó mediante la Laplaciana de la Gaussiana (LoG), existiendo otros preprocesamientos más complejos (Hirschmuller and Scharstein, 2007), pero que implican condiciones de flujo en el código no recomendables para su implementación en CUDA.

En la segunda etapa se calcula el coste de agregación mediante ventanas cuadradas. Aunque existen implementaciones más precisas (Gong et al., 2007), el tiempo de cómputo aumenta, siendo el resultado que aquí se presenta suficiente para conseguir un mapa denso de disparidad para zonas de la imagen con poca textura (calzada) o con patrones repetitivos utilizando ventanas cuadradas. En la tercera etapa se procede al cálculo de la disparidad mediante la utilización del método local WTA (Winner-Take-All) donde, para la posterior etapa de refinamiento, se ha construido el mapa de disparidad para la imagen izquierda (mapa izquierdo Fig. 7(c)) y para la imagen derecha (mapa derecho Fig. 7(d)) evitando cálculos redundantes al utilizar los del mapa izquierdo para obtener los del derecho. La última etapa es el refinamiento de la disparidad, implementándose un *cross-checking* con objeto de reducir los posibles errores en el mapa de disparidad (Fig. 7(b)) que suelen producirse en zonas donde no existe textura suficiente, siendo un ejemplo muy claro de ello el cielo (Fig. 7(a)). Otra fuente importante de errores son las zonas donde se producen profundas discontinuidades en la disparidad (Hirschmüller et al., 2002) y las áreas donde existen patrones repetidos, como pueden ser los edificios que se encuentran habitualmente en los entornos urbanos. No es muy relevante que no todos los errores sean eliminados en el *cross-checking*, debido a que el $u-v$ disparity se calcula mediante un proceso acumulativo, lo que le hace robusto ante la aparición

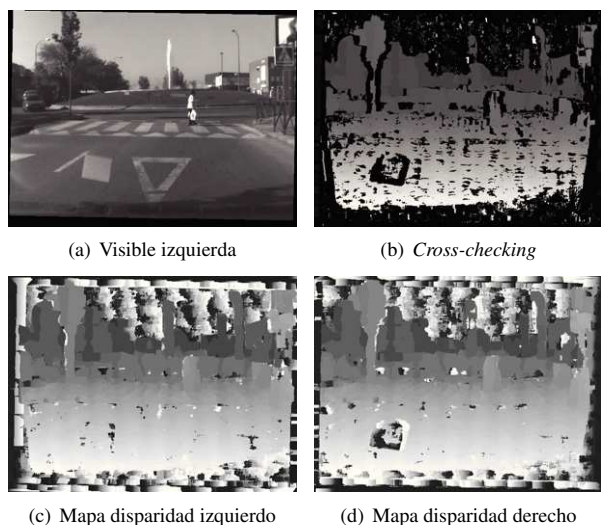


Figura 7: Ejemplo de la construcción del mapa denso de disparidad en entornos urbanos.

de una cierta cantidad de errores en la construcción del mapa de disparidad.

La implementación que aquí se presenta se basa en el trabajo presentado por (Stam, 2008), al cual se ha añadido: el pre-procesamiento usando la LoG (Laplaciana de la Gausiana), la obtención de ambos mapas de disparidad (izquierdo y derecho) evitando realizar cálculos redundantes, el *cross-checking* y la generación del *u-v disparity*. En la implementación se han utilizado imágenes de tamaño 640x480 en niveles de gris. El algoritmo ha sido probado en dos GPUs diferentes. La primera de ellas es una NVIDIA Geforce 9300 GS, típica de ordenadores portátiles y una segunda más potente, la NVIDIA Quadro FX 380 LP.

4.1.1. Procesamiento mediante LoG

La primera tarea es la construcción de la LoG sobre ambas imágenes. Las dos imágenes de entrada son almacenadas en la memoria de texturas de la tarjeta gráfica, siendo esta memoria de solo lectura y, en consecuencia, más rápida que el resto. Además puede ser direccionada usando coordenadas matriciales como las de la imagen. La implementación más sencilla para la obtención de la LoG emplea un hilo para cada píxel de la imagen, que lee los datos necesarios de la memoria de texturas y computa la convolución con el filtro, accediendo así varios hilos a las mismas zonas de memoria. Para reducir estos conflictos de acceso, la convolución se obtiene siguiendo los tres pasos siguientes:

1. Se emplea un hilo para cada píxel, aunque ahora este hilo calcula primero la convolución de la fila correspondiente a su píxel con cada una de las filas del filtro y entonces almacena los resultados en la memoria compartida. En este caso el filtro es simétrico y, por tanto, se pueden evitar cálculos redundantes.
2. Para que todos los resultados parciales del paso anterior se hayan almacenado, antes de continuar se impone una

sincronización.

3. Cuando todos los hilos se han sincronizado, cada uno calcula el valor final de la convolución para su píxel correspondiente, sumando los resultados parciales de la convolución de las filas vecinas anteriormente almacenadas en la memoria compartida.

4.1.2. Cálculo de la Función de Coste

Una vez han sido obtenidas y almacenadas ambas LoG en la memoria de texturas, se puede pasar a calcular la función de coste SD, almacenando el resultado en una matriz. Esta operación y las restantes hasta la etapa de refinamiento, deben repetirse para cada posible valor de disparidad, desplazando las LoG a lo largo del eje u . Como muestra la Fig. 8(a), se utiliza un hilo para cada columna de píxeles, obteniendo la SD resultante de ambas imágenes para un valor determinado de disparidad (desplazamiento).

4.1.3. Cálculo del Coste de Agregación

El coste de agregación se calcula fila a fila y en dos etapas: en primer lugar, se usa un hilo para sumar el SD para cada columna de la ventana de agregación y se almacena el resultado en la memoria compartida, lo que permite que estén accesibles para el resto de hilos. En segundo lugar, se utiliza un único hilo para obtener el valor final del coste de agregación para cada píxel, sumando el coste de agregación acumulado de cada una de las columnas vecinas dentro de la ventana de agregación correspondiente al píxel. La ventaja de proceder de esta manera es que el coste acumulado de las columnas puede ser reutilizado para obtener el coste de agregación para varios píxeles. Se presenta un esquema en la Fig. 8(b).

Una vez que todos los píxeles de una fila han sido procesados, la siguiente fila se calcula utilizando el concepto presentado en (Faugeras et al., 1993), que consiste en sustraer del coste de agregación de la fila anterior, el coste de agregación correspondiente a la primera fila de la ventana de agregación anterior y añadir el coste de agregación de la última fila de la nueva ventana de agregación. De esta manera se ahorra una ingente cantidad de cálculos redundantes.

4.1.4. Cálculo de la Disparidad

Cuando el coste de agregación ha sido calculado para cada píxel con un valor concreto de disparidad (desplazamiento), el hilo verifica si este nuevo valor de coste de agregación es el mínimo y, en ese caso, almacena este valor como el nuevo valor mínimo de coste de agregación, así como el valor de disparidad asociado a éste en dos matrices situadas en la memoria global para posteriores comparaciones. A fin de evitar cálculos innecesarios, el coste de agregación con coordenadas (u,v) en el mapa izquierdo es el mismo coste de agregación con coordenadas $(u-d,v)$ para el mapa derecho. El tiempo de cómputo se ha reducido en aproximadamente un 35 % frente a la implementación original de Stam al utilizar esta optimización.

4.1.5. Refinamiento de la Disparidad

Los pasos previos deben repetirse para cada posible valor de disparidad y, cuando se han obtenido los mejores valores de

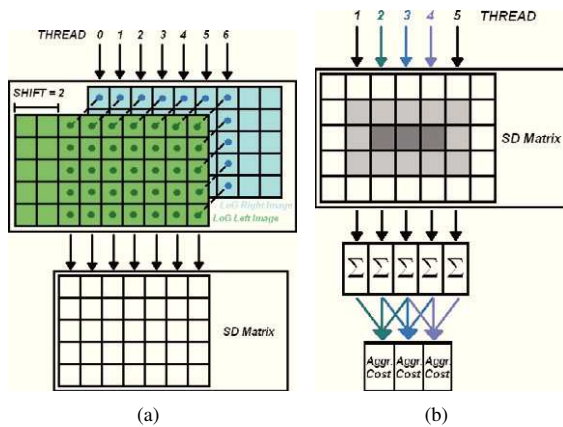


Figura 8: Esquemas de la implementación del mapa de disparidad con CUDA. (a) Esquema cálculo func. de coste. (b) Esquema del cálculo coste de agregación.

disparidad para cada mapa (izquierdo y derecho), ya se puede realizar el *cross-checking* entre ambos mapas. La implementación consiste en usar un hilo por cada píxel que compare los valores de cada mapa, si estos coinciden se conserva su valor y, en caso contrario, se fija a 0 el valor de ese píxel.

4.1.6. *u-v disparity*

Como se ha comentado anteriormente, para obtener el *u-disparity* hay que realizar el histograma a cada columna del mapa de disparidad y, en el caso del *v-disparity*, a cada fila. A tal efecto se va a utilizar un hilo para cada columna o fila. Este hilo calcula el respectivo histograma y cuando ha finalizado, comprueba la aparición de posibles desbordamientos, almacenando el resultado en la memoria global. En el caso del *u-disparity* se desea umbralizarlo para detectar los obstáculos, de tal modo que en lugar de comprobar los posibles desbordamientos se comprueban qué píxeles del mismo superan el umbral de altura de los obstáculos.

4.1.7. *Mapa de Obstáculos y Mapa Libre*

Cuando el *u-disparity* ha sido umbralizado, se utiliza un hilo para cada píxel umbralizado cuyas coordenadas son (u, d) . Cada hilo recorre la columna (u) correspondiente del mapa de la disparidad, conservando aquellos píxeles que cumplen que su nivel de gris es igual a (d) en caso de construir el mapa de obstáculos y conservando aquellos que cumplen que su nivel de gris es diferente a (d) en el caso del mapa libre.

4.1.8. *Análisis de Resultados*

Se han realizado diversos ensayos variando el valor máximo de disparidad y el tamaño de la ventana de agregación, pues estas dos variables de configuración tienen una gran influencia en el tiempo de cómputo. Se presenta un resumen de los resultados usando la tarjeta gráfica GeForce 9300 GS, en la Tabla 1.

Tanto la disparidad máxima como el tamaño de la ventana de agregación incrementan el tiempo de cómputo, pero resulta interesante que el aumento del valor de disparidad máxima afecta en mayor medida que el incremento del tamaño de la

ventana de agregación. Esto es debido a que la implementación para el cálculo del coste de agregación anteriormente descrita reduce los cálculos redundantes. Por ejemplo, para un incremento desde un tamaño de ventana de 11×11 a 21×21 implica aproximadamente 2.5 veces más píxeles a ser procesados y produce únicamente un incremento de 0.1 veces aproximadamente en el tiempo de cómputo.

Tabla 1: Tiempos de cómputo para la construcción del mapa de disparidad y del *u-v disparity* para diferentes tamaños de la ventana de agregación y de la disparidad máxima usando la NVIDIA GeForce 9300 GS.

Tamaño ventana	11x11	15x15	21x21
Disparidad Max			
30	0.114 s	0.118 s	0.125 s
40	0.144 s	0.150 s	0.159 s
60	0.203 s	0.212 s	0.225 s

La tabla 2 presenta los tiempos de cómputo para cada una de las etapas de la construcción, tanto del mapa de disparidad como del *u-v disparity* utilizando las dos tarjetas gráficas de las que se dispone. La configuración final se ha fijado en un valor de 30 para la disparidad máxima y un tamaño de ventana de agregación de 17×17 . Usando la tarjeta gráfica más potente de las dos, es posible computar en tiempo real con un *frame rate* de 20 imágenes por segundo.

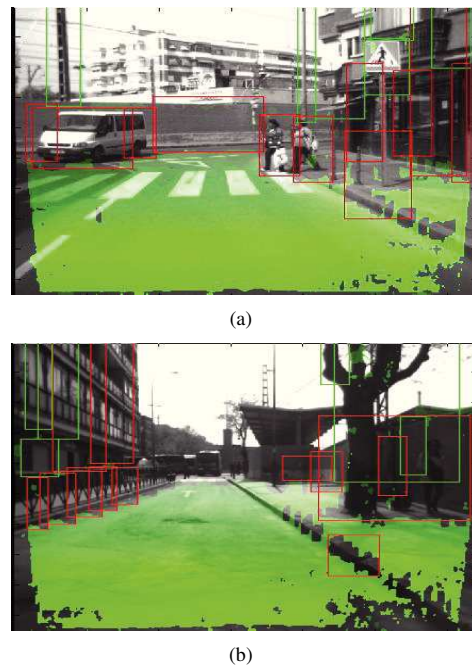


Figura 9: Ejemplos del resultado del algoritmo en entornos urbanos. Las zonas transitables aparecen en verde, los obstáculos no elevados recuadrados en rojo y en verde los elevados.

5. Resultados Experimentales

En esta sección se presentan los resultados de varias pruebas realizadas en entornos reales de circulación, y que evidencian

Tabla 2: Tiempos de cómputo para cada etapa del mapa de disparidad y del u-v disparity para dos tarjetas gráficas diferentes.

Tiempo Proc.(ms)	LoG	Mapa Disp.	Cross Cheking	u disp	v disp	TOTAL
GeForce 9300GS	6,37	107,5	1,72	0,94	1,85	119,7 (ms)
Quadro FX 380 LP	2,85	43,3	0,38	0,89	1,88	49,9 (ms)

la potencia del algoritmo (Fig. 9). En primer lugar, se detalla un experimento para evaluar el cálculo del ángulo de cabeceo (α) cuando el vehículo circula por un badén (en trabajos futuros se estudiará la variación del ángulo de cabeceo para llegar a determinar cuándo el vehículo se aproxima a subidas o bajadas pronunciadas o se produce una frenada de emergencia). A continuación, se presentan las localizaciones de peatones que describen dos trayectorias: un zig-zag delante del vehículo y un cruce entre ellos.

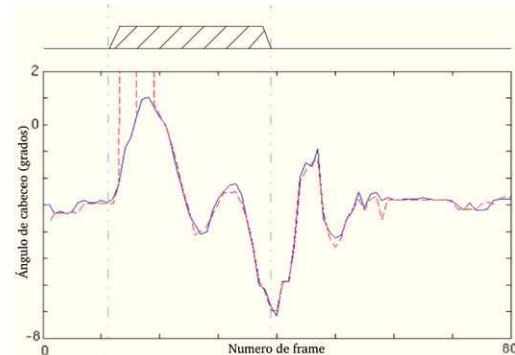
5.1. Badenes

Tal como se indicó anteriormente, una vez que se ha obtenido el perfil de la calzada, es posible calcular el ángulo de cabeceo (α) entre el sistema estéreo y la calzada. Se ha realizado un experimento para evaluar el cálculo de este ángulo (α) por medio del perfil de la calzada y consiste en calcular dicho ángulo para cada *frame* de una secuencia donde aparece un badén (Fig. 10(b)). Las dimensiones del badén son aproximadamente de 0.2 m. de alto por 3 m. de largo.

La Fig. 10(a) muestra un esquema de la calzada y en la parte inferior, un gráfico del ángulo de cabeceo (α) a lo largo de la misma, donde la línea azul continua representa los valores del ángulo obtenidos usando el mapa libre y la línea roja discontinua, los valores usando el mapa de disparidad. No hay diferencias reseñables entre ambas líneas salvo al comienzo del badén, donde el ángulo calculado usando el mapa de disparidad (línea roja) es erróneo en diversos *frames*. Es importante destacar que las oscilaciones del ángulo de cabeceo al principio y al final del badén se deben a la suspensión del vehículo. El estudio del ángulo de cabeceo puede ser extensible a otros casos, tales como baches o frenazos repentinos.

5.2. Detección y Localización de Obstáculos

Se han realizado diversos experimentos en entornos urbanos a fin de evaluar lo robustas y fiables que son la detección y la localización propuesta. Con objeto de localizar únicamente a los peatones que aparecen en la secuencia, se ha utilizado un clasificador de peatones aplicado sobre las regiones de interés obtenidas en la etapa de detección de obstáculos. Se presenta un experimento específico que puede dividirse en dos estadios: en primer lugar, un peatón que sigue una trayectoria en zig-zag y después dos peatones cruzándose delante del vehículo. La Fig. 11(a) y la Fig. 11(c) muestran los resultados de la clasificación de obstáculos para cuatro imágenes en cada estadio; los peatones aparecen en azul, los obstáculos elevados en verde y finalmente, los que están sobre el suelo que no son peatones aparecen en rojo. La Fig. 11(b) y la Fig. 11(d) presentan los resultados de la localización de los peatones en ambos estadios, los círculos azules representan la localización obtenida por medio de la ecuaciones (6) y las cruces rojas representan las



(a)



(b)

Figura 10: (a) Esquema de la calzada donde aparece un badén y en la parte inferior el gráfico del ángulo de cabeceo a lo largo de la calzada. (b) Un *frame* de la secuencia donde aparece el badén.

localizaciones obtenidas por medio de la ecuaciones (5). Como se puede apreciar, el uso del perfil de la calzada (2) produce una mejora de un orden de magnitud en la resolución de la localización de obstáculos, siendo más significativa en áreas alejadas del vehículo. Con respecto a la clasificación de peatones, el sistema ha clasificado correctamente a estos en un 78 % de los casos, con un 4 % de falsos positivos. Es importante destacar que los peatones han sido detectados como un obstáculo en el 97 % de los casos.

El *frame rate* del sistema es superior a 10 imágenes por segundo debido a que buena parte del algoritmo se procesa en la GPU a través de NVIDIA CUDA, lo cual permite reducir significativamente el tiempo de cómputo con respecto al procesado en CPU. Por ejemplo, la construcción del mapa de disparidad es una tarea costosa: la misma implementación para generar el mapa de disparidad es 15 veces más rápida usando la GPU que la CPU. Para realizar esta comparación se ha empleado un Core2Duo 2Ghz y 1GB de RAM como CPU y una NVIDIA Quadro FX 380 LP como GPU.

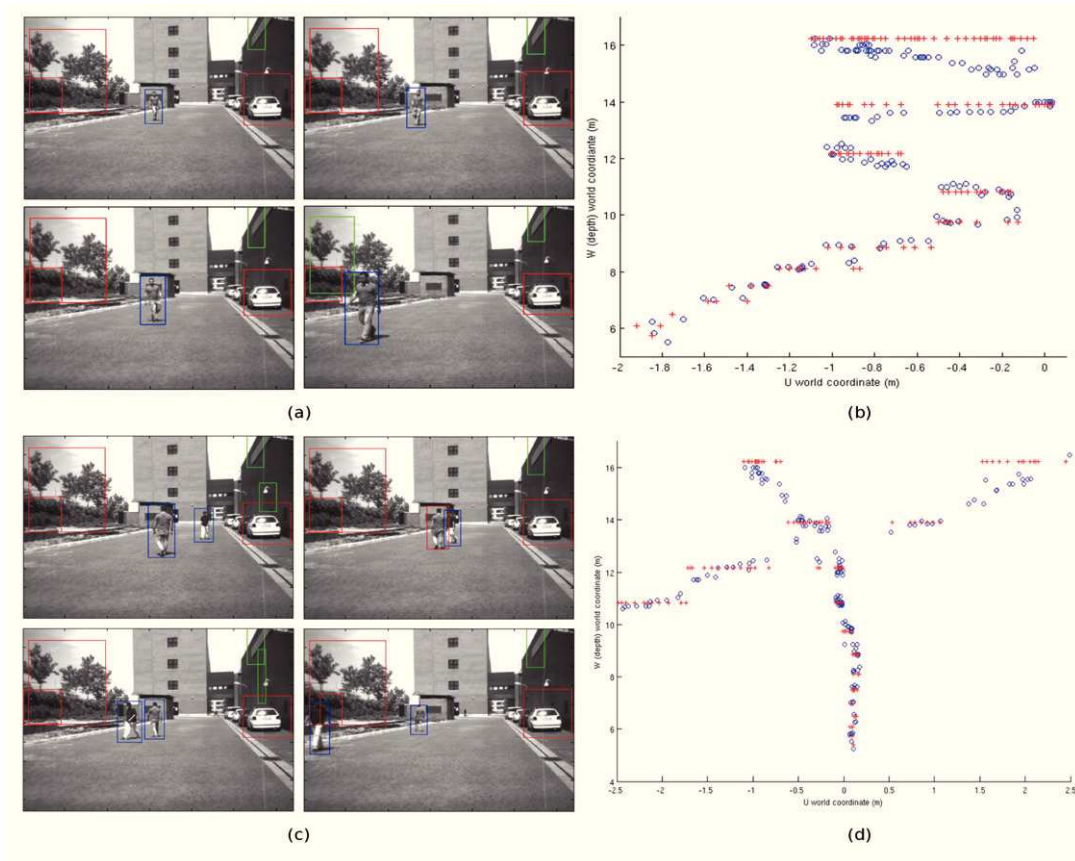


Figura 11: Experimento de clasificación de peatones y localización en un entorno urbano. (a) y (c) Cuatro imágenes de cada estadio (zig-zag y cruce) donde las regiones de interés aparecen sobre los obstáculos. (b) y (d) Localizaciones de peatones como resultado de las trayectorias que siguen.

6. Conclusiones

Se ha presentado un sistema para la detección de obstáculos y zonas transitables basado en visión estereó enfocada a los sistemas avanzados de ayuda a la conducción, pero cuyo uso también es posible en vehículos autónomos. Los entornos urbanos son zonas muy complejas para este tipo de sistemas, así pues, se detalla el resultado de un estudio de los casos más relevantes en los que se suelen producir funcionamientos erróneos y sus posibles soluciones. Se ha expuesto una nueva clasificación de obstáculos que permite distinguir entre aquellos que están elevados y los que se encuentran sobre la calzada desarrollándose, para este último grupo, una nueva formulación para su localización por medio del perfil de la calzada, que mejora en gran medida la resolución de esta localización. Para evaluar el algoritmo implementado se han realizado diferentes pruebas en entornos urbanos, presentándose los resultados de una selección de los mismos, que valida la eficacia de las distintas partes del algoritmo.

English Summary

Obstacle detection and free spaces in urban environments for advanced driver assistance systems based on stereo vi-

sion algorithms implemented in GPU.

Abstract

Both advanced driver assistance systems (ADAS) applied to the improvement of road safety, and autonomous navigation vehicle systems require more and more complex sensors and algorithms capable of obtaining and interpreting the information of the road environment. The greatest difficulties arise in analysing the information of the urban environments, because of the large number of elements which have different characteristics in urban areas. These systems require to interpret the information in real time to improve the decision-making. On the other hand, the stereo vision is usable in modeling systems because of the great amount of information that it provides, but at the same time, the algorithms based on this technique have a high computation time which makes difficult its implementation in real time applications. This paper presents an algorithm based on stereo vision for detecting obstacles and free spaces in urban environments and it has been implemented principally in GPU (Graphic Processing Unit) to reduce the computation time and achieving that it works in real time.

Keywords:

Computer Vision, Autonomous Vehicles, Detection Algorithms, Real time systems.

Agradecimientos

Este trabajo ha sido subvencionado por la CICYT a través del proyecto FEDORA (TRA2010- 20225-C03-01) y el proyecto Driver Distraction Detector System (TRA2011-29454-C03-02).

Referencias

- Armingol, J., de la Escalera, A., Hilario, C., Collado, J., Carrasco, J., Flores, M., Pastor, J., Rodriguez, F., 2007. Ivvi: Intelligent vehicle based on visual information. *Robotics and Autonomous Systems* 55 (12), 904–916.
- Bertozzi, M., Broggi, A., Grisleri, P., Graf, T., Meinecke, M., 2003. Pedestrian detection in infrared images. In: *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE. IEEE*, pp. 662–667.
- Broggi, A., Cappalunga, A., Caraffi, C., Cattani, S., Ghidoni, S., Grisleri, P., Porta, P., Posterli, M., Zani, P., Beck, J., 2008. The passive sensing suite of the terramax autonomous vehicle. In: *Intelligent Vehicles Symposium, 2008 IEEE. IEEE*, pp. 769–774.
- Broggi, A., Caraffi, C., Fedriga, R., Grisleri, P., 2005. Obstacle detection with stereo vision for off-road vehicle navigation. In: *Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on. IEEE*, pp. 65–65.
- Brown, M., Burschka, D., Hager, G., 2003. Advances in computational stereo. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 25 (8), 993–1008.
- Cabani, I., Toulminet, G., Bensrhair, A., 2006. A color stereo vision system for extraction of 3d edges of obstacle. In: *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE. IEEE*, pp. 307–312.
- Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 1. Ieee*, pp. 886–893.
- Enzweiler, M., Gavrila, D., 2009. Monocular pedestrian detection: Survey and experiments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31 (12), 2179–2195.
- Faugeras, O., Viéville, T., Theron, E., Vuillemin, J., Hotz, B., Zhang, Z., Moll, L., Bertin, P., Mathieu, H., Fua, P., et al., 1993. Real-time correlation-based stereo: algorithm, implementations and applications.
- Flores, M., Armingol, J., de la Escalera, A., 2011. Sistema avanzado de asistencia a la conducción para la detección de la somnolencia. *Revista Iberoamericana de Automática e Informática Industrial RIAI* 8 (3), 216–228.
- Fusiello, A., Trucco, E., Verri, A., 2000. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications* 12 (1), 16–22.
- Gong, M., Yang, R., Wang, L., Gong, M., 2007. A performance study on different cost aggregation approaches used in real-time stereo matching. *International Journal of Computer Vision* 75 (2), 283–296.
- Guo, C., Mita, S., McAllester, D., 2009. Drivable road region detection using homography estimation and efficient belief propagation with coordinate descent optimization. In: *Intelligent Vehicles Symposium, 2009 IEEE. IEEE*, pp. 317–323.
- Hautié, N., Labayrade, R., Perrollaz, M., Aubert, D., 2006. Road scene analysis by stereovision: a robust and quasi-dense approach. In: *Control, Automation, Robotics and Vision, 2006. ICARCV'06. 9th International Conference on. IEEE*, pp. 1–6.
- Hirschmüller, H., Innocent, P., Garibaldi, J., 2002. Real-time correlation-based stereo vision with reduced border errors. *International Journal of Computer Vision* 47 (1), 229–246.
- Hirschmüller, H., Scharstein, D., 2007. Evaluation of cost functions for stereo matching. In: *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on. IEEE*, pp. 1–8.
- Hu, Z., Lamosa, F., Uchimura, K., 2005. A complete uv-disparity study for stereovision based 3d driving environment analysis. In: *3-D Digital Imaging and Modeling, 2005. 3DIM 2005. Fifth International Conference on. IEEE*, pp. 204–211.
- Hu, Z., Uchimura, K., 2005. Uv-disparity: an efficient algorithm for stereovision based scene analysis. In: *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE. IEEE*, pp. 48–54.
- Kim, J., Hwangbo, M., Kanade, T., 2009. Parallel algorithms to a parallel hardware: Designing vision algorithms for a gpu. In: *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on. IEEE*, pp. 862–869.
- Labayrade, R., Aubert, D., 2003a. In-vehicle obstacles detection and characterization by stereovision. *Proc. IEEE In-Vehicle Cognitive Comput. Vis. Syst.*, 1–3.
- Labayrade, R., Aubert, D., 2003b. A single framework for vehicle roll, pitch, yaw estimation and obstacles detection by stereovision. In: *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE. IEEE*, pp. 31–36.
- Labayrade, R., Aubert, D., Tarel, J., 2002. Real time obstacle detection in stereovision on non flat road geometry through v-disparity representation. In: *Intelligent Vehicle Symposium, 2002. IEEE. Vol. 2. IEEE*, pp. 646–651.
- Lee, C., Lim, Y., Kwon, S., Lee, J., 2008. Obstacle localization with a binarized v-disparity map using local maximum frequency values in stereo vision. In: *Signals, Circuits and Systems, 2008. SCS 2008. 2nd International Conference on. IEEE*, pp. 1–4.
- Lemonde, V., Devy, M., 2004. Obstacle detection with stereovision. *Mechatronics & Robotics (MECHROB'04)* 3, 919–924.
- Ling, B., Zeifman, M., Gibson, D., 2007. Multiple pedestrian detection using ir led stereo camera. In: *Proceedings of SPIE. Vol. 6764. p. 67640A*.
- Musleh, B., García, F., Otamendi, J., Armingol, J., De la Escalera, A., 2010. Identifying and tracking pedestrians based on sensor fusion and motion stability predictions. *Sensors* 10 (9), 8028–8053.
- Nvidia, C., 2008. Programming guide.
- NVIDIA, C., 2009. C programming best practices guide. *Cuda Toolkit 2*.
- Olmeda, D., de la Escalera, A., Armingol, J., 2011. Far infrared pedestrian detection and tracking for night driving. *Robotica* 29 (04), 495–505.
- Pérez, J., Milanés, V., Alonso, J., Onieva, E., De Pedro, T., 2010. Adelantamiento con vehículos autónomos en carreteras de doble sentido. *Revista Iberoamericana de Automática e Informática Industrial RIAI* 7 (3), 25–33.
- Petrovskaya, A., Thrun, S., 2009. Model based vehicle detection and tracking for autonomous urban driving. *Autonomous Robots* 26 (2), 123–139.
- Premebida, C., Ludwig, O., Nunes, U., 2009. Exploiting lidar-based features on pedestrian detection in urban scenarios. In: *Intelligent Transportation Systems, 2009. ITSC'09. 12th International IEEE Conference on. IEEE*, pp. 1–6.
- Scharstein, D., Szeliski, R., 2002. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision* 47 (1), 7–42.
- Soquet, N., Aubert, D., Perrollaz, M., et al., 2007. Free space estimation for autonomous navigation.
- Stam, J., 2008. Stereo imaging with cuda. *OpenVIDIA*, january.
- Sun, Z., Bebis, G., Miller, R., 2006. Monocular precrash vehicle detection: features and classifiers. *Image Processing, IEEE Transactions on* 15 (7), 2019–2034.
- Taylor, T., Geva, S., Boles, W., 2004. Monocular vision as a range sensor. *International Conference on Computational Intelligence for Modelling, Control and Automation*.
- Thrun, S., 2003. Learning occupancy grid maps with forward sensor models. *Autonomous robots* 15 (2), 111–127.
- Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M., Dolan, J., Duggins, D., Galatali, T., Geyer, C., et al., 2008. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics* 25 (8), 425–466.
- Wang, L., Gong, M., Gong, M., Yang, R., 2006. How far can we go with local optimization in real-time stereo matching. In: *3D Data Processing, Visualization, and Transmission, Third International Symposium on. IEEE*, pp. 129–136.
- Xu, Y., Zhao, M., Wang, X., Zhang, Y., Peng, Y., Yuan, Y., Liu, H., 2009. A method of stereo obstacle detection based on image symmetrical move. In: *Intelligent Vehicles Symposium, 2009 IEEE. IEEE*, pp. 36–41.
- Zhao, J., Whitty, M., Katupitiya, J., 2009. Detection of non flat ground surfaces using v disparity images. In: *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on. IEEE*, pp. 4584–4589.