

# Estabilización de imágenes obtenidas por una cámara a bordo de un vehículo en movimiento para la mejora de un sistema Light-Beam

Álvaro González, Luis M. Bergasa, Pablo F. Alcantarilla, Pedro Jiménez

**Abstract**—Los sistemas de visión artificial juegan un importante papel en numerosos Sistemas de Transporte Inteligente (STI), tales como monitorización del tráfico, asistencia al conductor o conducción automática de vehículos. Uno de los principales problemas que suelen sufrir este tipo de sistemas, bien estén instalados en puntos fijos del entorno o en los propios vehículos, es la inestabilidad en las imágenes. En este documento, se presenta una técnica de estabilización de imágenes obtenidas por una cámara a bordo de un vehículo en movimiento, con el objetivo de garantizar el correcto funcionamiento de un STI conocido como Light-Beam. Esta técnica permite estimar movimientos de la cámara en el mundo 3D únicamente a partir de las líneas de la carretera y del punto de fuga del frame bajo estudio, en combinación con la información extraída en el frame anterior. Tras una fase de suavizado del movimiento, se obtiene la imagen estabilizada aplicando una transformación geométrica. Se han realizado una serie de experimentos con datos artificiales y reales y los resultados obtenidos han demostrado la eficiencia de la técnica propuesta.

**Index Terms**—Estabilización de imágenes, estimación del movimiento, compensación de la imagen, Light-Beam.

## I. INTRODUCCIÓN

EL grupo de Robótica de servicios y Tecnologías para la seguridad vial del Departamento de Electrónica de la Universidad de Alcalá ha colaborado en el desarrollo de un Sistema de Transporte Inteligente (STI) conocido como Light-Beam o “Haz Inteligente” para la empresa FICOMIRRORS S.A [1]. Se trata de un sistema de control del haz de iluminación de un vehículo aplicable en carreteras delimitadas con líneas y en ambientes nocturnos o de poca luz, en función de los vehículos que circulan en el propio carril o en el carril contrario, con objeto de maximizar la zona iluminada sin deslumbrar al resto de conductores. Este sistema es capaz de detectar las luces de los vehículos que circulan por el mismo carril o por el contrario, discriminando otro tipo de luces del entorno, y calculando la distancia a la que se encuentran en todo momento con respecto al coche sobre el que está montado el sistema sensorial. En función de esa información es posible controlar de forma automática el haz de iluminación del vehículo, conmutando entre luces cortas y largas, de forma que el conductor pueda desentenderse de esta ardua tarea y dedicarse por completo a la conducción. De esta forma, se evitan deslumbramientos a los automóviles que circulan a una distancia muy cercana al nuestro u otras situaciones de peligro

y se maximiza la zona visible del vehículo propio cuando no aparecen otros a los que se pueda deslumbrar.

Para estimar la distancia a la que se encuentran el resto de vehículos del entorno usando visión monocular, se aplica un modelo en perspectiva de la cámara conocido como *pinhole*, de forma que un vehículo a una distancia  $Z$  de la cámara se proyectará en el plano imagen en un punto con coordenadas horizontal y vertical  $(u, v)$  respectivamente. Sin embargo, esta estimación de distancias se va a ver afectada por la inestabilidad en las imágenes, puesto que este efecto provoca que la proyección del vehículo sobre el plano imagen, y por consiguiente el cálculo de distancias, sea muy variable de un frame a otro. Esto da lugar a estimaciones con mucha incertidumbre y a que el sistema no funcione correctamente. Un ejemplo puede verse en la Fig. 1, en la que se muestra la estimación de distancia en comparación con la distancia real, en función del frame, para un vehículo que se aproxima en sentido contrario. Obsérvense las grandes desviaciones que sufre la distancia estimada respecto a la real. Además, la inestabilidad en las imágenes es un problema que tiene un gran impacto en la mayoría de los STI que emplean técnicas de visión artificial, como aquellos que detectan señales, peatones, vehículos u otros obstáculos. Por este motivo, el algoritmo de estabilización propuesto, además de ser necesario para garantizar el correcto funcionamiento del sistema Light-Beam, puede ser de interés para otro tipo de sistemas.

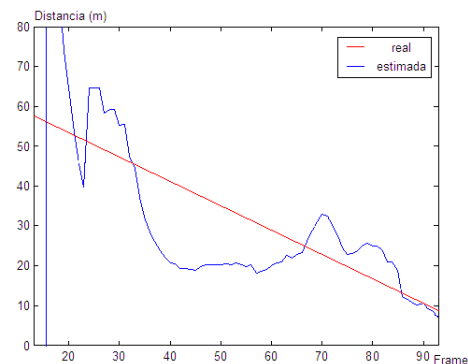


Fig. 1. Distancia de un vehículo que se aproxima en sentido contrario en función del frame

## II. ESTABILIZACIÓN DE IMÁGENES

En esta sección se describe la forma de abordar el problema. El algoritmo se tratará más detalladamente en secciones posteriores.

La cámara está situada en la parte frontal dentro del vehículo, como se muestra en la Fig. 2. Existen una serie de parámetros que son conocidos de antemano, como el pitch  $\beta_0$  y el yaw  $\gamma_0$  iniciales de la cámara, junto con la altura  $h$  a la que se encuentra respecto del suelo y la distancia focal  $f$  de la cámara. La cámara puede sufrir una serie de movimientos en el mundo 3D como los que se muestran en la Fig. 3. Estos movimientos son los que hay que estimar y suavizar para lograr la deseada estabilización.

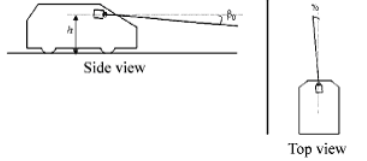


Fig. 2. Configuración de la cámara

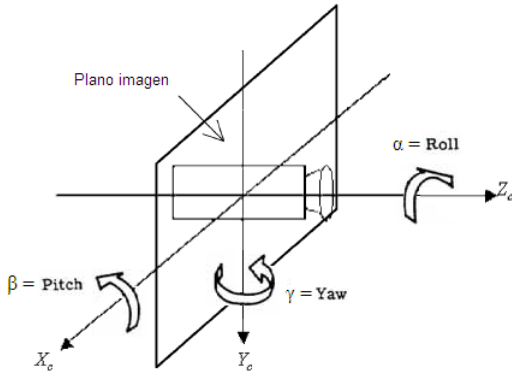


Fig. 3. Sistema de coordenadas de la cámara y movimientos 3D

La técnica de estabilización se divide en cuatro fases (veáse Fig. 4): extracción de características globales de la imagen, estimación del movimiento de la cámara, suavizado del movimiento y compensación de la imagen. Para cada frame de la secuencia de entrada, se extraen las líneas que delimitan el carril por el que circula el vehículo junto con el punto de fuga. Las denominamos características globales porque caracterizan a la carretera, el objeto más importante en una imagen de tráfico. Posteriormente, se estiman una serie de parámetros de movimientos de la cámara en el mundo 3D a partir de las características globales junto con la información extraída en el frame previo. Estos parámetros son la distancia transversal  $T_x$  de la cámara respecto al centro del carril y los ángulos  $\alpha$ ,  $\beta$  y  $\gamma$  de la Fig. 3. Estos parámetros se suavizan en la siguiente etapa, mediante un filtro de Kalman [3]. Por último, se obtiene la imagen estabilizada aplicando una transformación geométrica a la imagen de entrada.

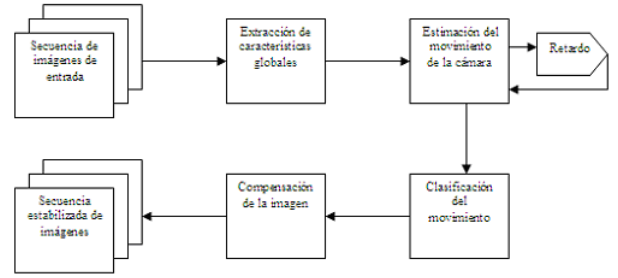


Fig. 4. Diagrama de bloques

### III. EXTRACCIÓN DE CARACTERÍSTICAS GLOBALES

En esta sección se describe la fase de extracción de las líneas de la carretera y del punto de fuga de la imagen de entrada. Las líneas de la carretera cumplen dos propiedades fundamentales: son de color blanco o amarillo y son paralelas en el mundo 3D, aunque sus proyecciones en el plano imagen intersectan en un punto conocido como punto de fuga [4]. Estas propiedades, junto con la configuración de la cámara dentro del vehículo (Fig. 2), dan como resultado que las líneas presentes en la imagen se correspondan con puntos de mucho brillo en la mitad inferior de la imagen. A continuación se explica el algoritmo de detección de líneas utilizado.

Sea  $I$  la imagen de entrada. En primer lugar, se aplica un filtro de realce de bordes. El principal objetivo del filtro de realce es resaltar aquellas características de la imagen que, por causa del mecanismo de captación o por error, hayan quedado emborronadas en la imagen. El filtro de realce de bordes ha resultado ser muy útil en aquellos casos en los que la cámara ha sufrido un movimiento brusco entre dos frames, como ocurre cuando, por ejemplo, el vehículo sobre el que va montada la cámara pasa por un badén o un resalto en la carretera. La Fig. 5 muestra la diferencia entre aplicar y no aplicar un filtro de realce de bordes y su efecto a la hora de extraer las líneas en la imagen.

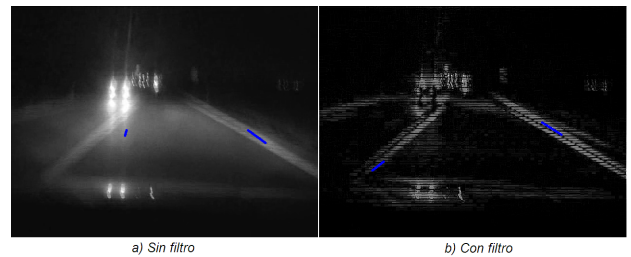


Fig. 5. Efecto de aplicar un filtro de realce de bordes

Para la implementación del realce se utilizan técnicas basadas en la aplicación de filtros. El realce de una imagen se basa en el filtro paso alto espacial, el cual elimina los componentes bajos de frecuencia y muestra sólo los detalles altos. La máscara de convolución usada comúnmente en el realce, es decir, la máscara de filtro paso alto, tiene un coeficiente positivo en su centro y coeficientes negativos alrededor del borde externo [5]. El inconveniente de este tipo de filtros

es que se reduce considerablemente el contraste global de la imagen, esto es, el número de grises presentes en la imagen original, como puede verse en la Fig. 5. Una opción para evitar este efecto es el uso de filtros Sharpen. Sin embargo, se ha comprobado que, para esta aplicación, es suficiente y funciona mucho mejor un filtro paso alto cuya máscara es la siguiente:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad (1)$$

Tras el realce de bordes, se aplica el algoritmo encargado de extraer las líneas de la imagen. Sea  $I_f$  la imagen resultante tras el proceso de realce e  $i_f(u, v)$  el valor del píxel  $(u, v)$ . Se definen los gradientes horizontales hacia la derecha y hacia la izquierda de dicho píxel como sigue:

$$\begin{aligned} d_{+m}(u, v) &= i_f(u, v) - i_f(u + m, v) \\ d_{-m}(u, v) &= i_f(u, v) - i_f(u - m, v) \end{aligned} \quad (2)$$

donde  $m$  especifica la anchura máxima en pixels de las líneas de la carretera. Se ha comprobado experimentalmente que el valor  $m = 15$  es el más adecuado. De esta forma, se construye una imagen  $J$  cuyos elementos se calculan como sigue:

$$j(u, v) = \begin{cases} d_{+m}(u, v) + d_{-m}(u, v) & , (d_{+m}(u, v) > 0) \text{ and } (d_{-m}(u, v) > 0) \\ 0 & , \text{ caso contrario} \end{cases} \quad (3)$$

La ecuación anterior toma un píxel de la imagen cuyo valor sea mayor que el de sus dos vecinos situados a unas distancias  $-m$  y  $m$  en horizontal. Un píxel que cumpla la condición anterior será un candidato a pertenecer a alguna de las líneas de la carretera. Posteriormente, se eliminan de  $J$  aquellos pixels con gradientes acumulados débiles, utilizando como umbral el valor medio  $t$  de todos los gradientes acumulados de aquellos pixels que cumplan la condición anterior. Se construye así una imagen binaria  $B$  en la que los pixels supervivientes toman un valor 1, y el resto 0.

$$b(u, v) = \begin{cases} 1 & \text{si } j(u, v) > t \\ 0 & \text{caso contrario} \end{cases} \quad (4)$$

Es importante resaltar que el algoritmo de extracción de líneas de la imagen sólo es necesario aplicarlo a la mitad inferior de la imagen, ya que es ahí donde se proyectan las líneas de la carretera. De esta forma, se consigue acelerar el proceso de búsqueda de líneas en la imagen. El resultado se muestra en la Fig. 6(b). Obsérvese que existe un ruido bastante significativo acompañando a las líneas de la carretera. Para eliminarlo se aplica un etiquetado de regiones con conectividad seguido de un filtrado por tamaño. El filtrado por tamaño elimina todas las regiones pequeñas, basándose en un umbral definido como el valor medio del número de píxeles en todas las regiones con conectividad. La Fig. 6(c) muestra el resultado de eliminar el ruido.

El siguiente paso consiste en localizar los puntos medios de las líneas, es decir, en realizar un proceso de esqueletización. Para ello, se recorre la imagen fila a fila desde la última. La búsqueda se realiza en dos regiones: el cuadrante inferior derecho y el cuadrante inferior izquierdo de la imagen, puesto que es donde se localizarán cada una de las líneas que

delimitan el carril. En cada fila, se buscan segmentos cuyas longitudes sean inferiores al valor  $m$  definido previamente. Por cada segmento que cumpla dicha condición, se marca el punto medio como candidato. Una vez finalizado este proceso, el siguiente paso consiste en detectar las líneas presentes y quedarnos con aquellas que se correspondan con las líneas de la carretera.

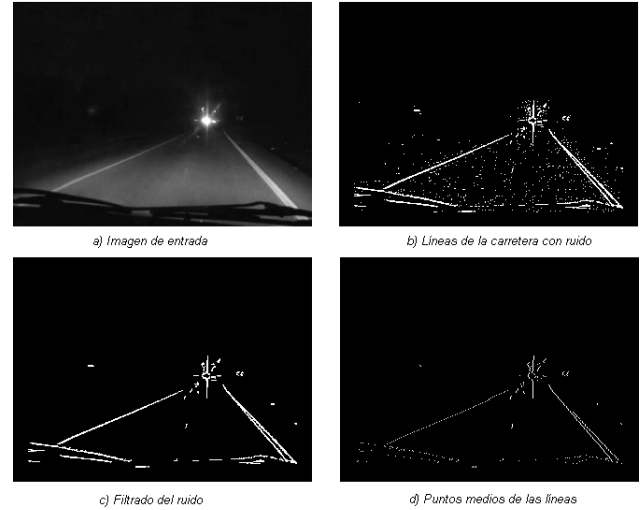


Fig. 6. Extracción de líneas en la imagen

Para la detección de las líneas se aplica la transformada de Hough, una técnica empleada en reconocimiento de patrones en imágenes. La transformada de Hough se aplica en dos regiones (Fig. 7(a)): el cuadrante inferior izquierdo y el cuadrante inferior derecho de la imagen, y se obtienen varias líneas (Fig. 7(b)). Para discriminar aquellas que no se corresponden con las líneas que delimitan el carril por el que circula el vehículo, se obtiene la proyección en la imagen de un modelo clotoidal de la carretera (Fig. 7(c)) [6]. Las líneas detectadas cuyas pendientes sean más parecidas a las pendientes de las líneas obtenidas mediante la proyección, son las líneas deseadas (Fig. 7(d)). Hay que señalar que el modelo clotoidal se actualiza en cada frame de la secuencia aplicando un filtro de Kalman, utilizando como vector de medidas la información extraída en el presente frame, es decir, los puntos  $(u_l, v_l)$  y  $(u_r, v_r)$  correspondientes a las líneas detectadas (véase [1]). Los valores de covarianza utilizados se han obtenido experimentalmente, aunque sus valores dependerán generalmente de la aplicación concreta sobre la que se emplee este algoritmo, y del tipo de carretera (más o menos bacheada) para esta aplicación en particular. En resumen, la proyección del modelo clotoidal para orientar la búsqueda de las líneas debe entenderse únicamente como una forma de aumentar la certeza de que las líneas elegidas se corresponden con las reales. Puesto que el modelo clotoidal se actualiza en cada frame con las medidas obtenidas y que entre dos frames consecutivos la imagen no va a variar mucho, la estimación de las líneas en cada frame es muy parecida a la realidad.

Una vez que se han detectado las líneas correspondientes a los límites de la carretera, el punto de fuga se calcula como la intersección de las mismas en el plano imagen (Fig. 8).

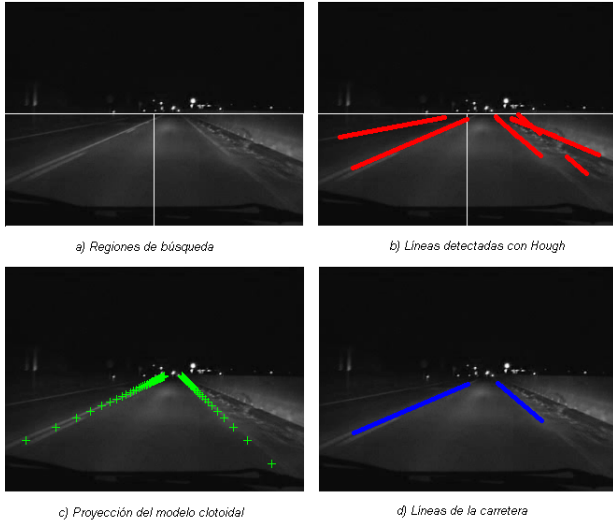


Fig. 7. Búsqueda de líneas



Fig. 8. Punto de fuga

#### IV. ESTIMACIÓN DEL MOVIMIENTO DE LA CÁMARA

En esta sección, se detalla el proceso seguido para estimar los distintos parámetros de movimiento de la cámara. En general, existen seis parámetros que definen un movimiento en el espacio 3D, tres de traslación ( $T_x, T_y, T_z$ ) y tres de rotación ( $\alpha, \beta, \gamma$ ). En nuestro caso, sólo nos interesan cuatro de ellos: los tres de rotación y uno de traslación,  $T_x$ . Se establecen dos sistemas de coordenadas. En primer lugar, un sistema de coordenadas global, como el mostrado en la Fig. 9, con el plano  $X_w - Y_w$  coincidente con el plano de la carretera, y el eje  $Z_w$  perpendicular al mismo. Este sistema se mueve a lo largo de la línea central del carril por el que circula el vehículo a la misma velocidad que lo hace éste.

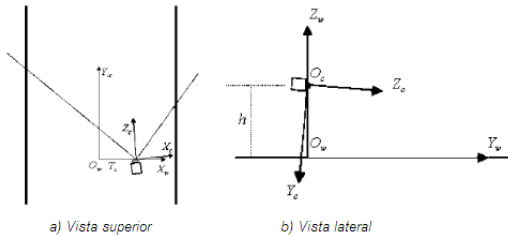


Fig. 9. Sistema de coordenadas global vs Sistema de coordenadas de la cámara

El segundo sistema de coordenadas es el de la cámara (Fig. 3), cuyo plano imagen coincide con el plano  $X_c - Y_c$  y el

eje  $Z_c$  es perpendicular al mismo. Típicamente se va a dar la situación mostrada en la Fig. 9. La cámara, además de estar situada a una altura  $h$  sobre el suelo, va a estar girada unos ciertos ángulos  $\alpha, \beta$  y  $\gamma$  y trasladada una distancia  $T_x$  respecto al centro del carril en la dirección transversal al movimiento del vehículo. Nuestro objetivo es estimar el valor de esos cuatro parámetros:  $\alpha, \beta, \gamma$  y  $T_x$ .

Para ello se va a seguir el siguiente proceso, cuya demostración puede encontrarse en [2]. En primer lugar, se calculan los parámetros  $\beta$  y  $\gamma$  a partir de la posición del punto de fuga en la imagen. Una vez calculados, se compensan sus efectos sobre las líneas de la carretera, y las líneas compensadas se emplean para calcular los parámetros  $T_x$  y  $\alpha$ .

##### A. Parámetros $\beta$ y $\gamma$

La relación entre las coordenadas en la imagen del punto de fuga ( $x_v, y_v$ ) y los parámetros de rotación  $\beta$  y  $\gamma$  es:

$$\begin{cases} x_v = f \cdot \tan \gamma \\ y_v = -f \cdot \frac{\tan \beta}{\cos \gamma} \end{cases} \quad (5)$$

Reordenando estas ecuaciones, se tiene lo siguiente:

$$\begin{cases} \gamma = \arctan \left( \frac{x_v}{f} \right) \\ \beta = -\arctan \left( \frac{y_v \cos \gamma}{f} \right) \end{cases} \quad (6)$$

siendo  $x_v, y_v$  y  $f$  conocidas.

Para compensar los efectos producidos por  $\beta$  y  $\gamma$  en las líneas de la carretera, rotamos la cámara alrededor de su eje  $Y_c$  según un ángulo  $-\gamma$  y posteriormente alrededor del nuevo eje  $X'_c$  por un ángulo  $-\beta$ .

$$\begin{aligned} R(-\gamma, -\beta) &= Rot(X'_c, -\beta) \cdot Rot(Y_c, -\gamma) = \\ &= \begin{bmatrix} \cos \gamma & 0 & -\sin \gamma \\ \sin \beta \sin \gamma & \cos \beta & \sin \beta \cos \gamma \\ \cos \beta \sin \gamma & -\sin \beta & \cos \beta \cos \gamma \end{bmatrix} \end{aligned} \quad (7)$$

Sean ( $x'_c, y'_c, z'_c$ ) las nuevas coordenadas después de la transformación anterior de un punto con coordenadas ( $x_c, y_c, z_c$ ). La relación entre ambas es la siguiente:

$$\begin{bmatrix} x'_c \\ y'_c \\ z'_c \end{bmatrix} = \begin{bmatrix} \cos \gamma & 0 & -\sin \gamma \\ \sin \beta \sin \gamma & \cos \beta & \sin \beta \cos \gamma \\ \cos \beta \sin \gamma & -\sin \beta & \cos \beta \cos \gamma \end{bmatrix} \cdot \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \quad (8)$$

Supóngase que ( $x_i, y_i$ ) y ( $x'_i, y'_i$ ) son las proyecciones en el plano imagen de los puntos anteriores. Según la expresión anterior y las expresiones de la proyección en perspectiva, la relación entre ambas es:

$$\begin{cases} x'_i = f \cdot \frac{x_i \cos \gamma - y_i \sin \gamma}{x_i \cos \beta \sin \gamma - y_i \sin \beta + f \cos \beta \cos \gamma} \\ y'_i = f \cdot \frac{x_i \sin \beta \sin \gamma + y_i \cos \beta + f \sin \beta \cos \gamma}{x_i \cos \beta \sin \gamma - y_i \sin \beta + f \cos \beta \cos \gamma} \end{cases} \quad (9)$$

Por tanto, dado un punto ( $x_i, y_i$ ) perteneciente a una de las líneas de la carretera, la expresión anterior dará su posición

en la imagen una vez compensada la rotación debida a  $\beta$  y  $\gamma$ . Además el punto de fuga se situará en el centro de la imagen  $(0,0)$ , puesto que se puede demostrar que la posición del punto de fuga en la imagen sólo depende de los parámetros  $\beta$  y  $\gamma$  [2].

### B. Parámetros $T_x$ y $\alpha$

Las líneas compensadas son necesarias para calcular los parámetros  $T_x$  y  $\alpha$ . Sea un punto  $(x_i, y_i)$  perteneciente a una de estas líneas compensadas. Supóngase que la cámara está trasladada  $T_x$  en el eje  $X_w$ , seguida de una rotación por un ángulo  $\alpha$  alrededor del eje  $Z'_c$  resultante, en el sistema de coordenadas de la cámara. Las nuevas coordenadas del punto son:

$$\begin{cases} x'_i = (x_i - dy_i) \cos \alpha - y_i \sin \alpha \\ y'_i = (x_i - dy_i) \sin \alpha + y_i \cos \alpha \end{cases} \quad (10)$$

siendo  $d = \frac{T_x}{h}$ . Para conocer  $T_x$ , en primer lugar hay que calcular  $d$ , el cual se determina de la siguiente manera.

Sean  $s$  y  $s'$  las pendientes de una de las líneas de la carretera antes y después de la transformación anterior (10) de la cámara. Como las coordenadas del punto de fuga son  $(0,0)$  tanto antes como después de dicha transformación, al no depender ni de  $T_x$  ni de  $\alpha$ , permaneciendo en el centro de la imagen,  $s$  y  $s'$  pueden calcularse así:

$$\begin{cases} s = \frac{0 - y_i}{0 - x_i} = \frac{y_i}{x_i} \\ s' = \frac{0 - y'_i}{0 - x'_i} = \frac{y'_i}{x'_i} \end{cases} \quad (11)$$

Sustituyendo  $(x'_i, y'_i)$  según la expresión (10), se tiene:

$$s' = \frac{y'_i}{x'_i} = \frac{(x_i - dy_i) \sin \alpha + y_i \cos \alpha}{(x_i - dy_i) \cos \alpha - y_i \sin \alpha} = \frac{(x_i - dy_i) \tan \alpha + y_i}{(x_i - dy_i) - y_i \tan \alpha} \quad (12)$$

Despejando  $\tan \alpha$  de la ecuación anterior, se obtiene:

$$\tan \alpha = \frac{s' (1 - ds) - s}{(1 - ds) + s' s} \quad (13)$$

Supóngase que  $l_{t-1}$  y  $r_{t-1}$  son las pendientes de las dos líneas que delimitan el carril por el que está circulando el vehículo, de la imagen inmediatamente anterior una vez compensadas todas las rotaciones, y  $l'_t$  y  $r'_t$  las pendientes de dichas líneas en el frame actual después de compensar la rotación por  $\beta$  y  $\gamma$ . Análogamente a la expresión anterior, se tiene lo siguiente:

$$\begin{cases} \tan \alpha = \frac{l'_t (1 - dl_{t-1}) - l_{t-1}}{(1 - dl_{t-1}) + l'_t l_{t-1}} \\ \tan \alpha = \frac{r'_t (1 - dr_{t-1}) - r_{t-1}}{(1 - dr_{t-1}) + r'_t r_{t-1}} \end{cases} \quad (14)$$

Igualando las dos ecuaciones, se obtiene una ecuación de segundo orden:

$$pd^2 + qd + r = 0 \quad (15)$$

donde:

$$\begin{cases} p = (l'_t - r'_t) l_{t-1} r_{t-1} \\ q = -(l'_t - r'_t) (l_{t-1} + r_{t-1}) \\ r = (l'_t - r'_t) (1 + l_{t-1} r_{t-1}) + (r_{t-1} - l_{t-1}) (1 + l'_t r'_t) \end{cases} \quad (16)$$

Resolviendo (15), se obtienen dos soluciones para  $d$ , de las que se elige aquella más próxima a la solución obtenida en el frame anterior. Una vez determinado  $d$ ,  $T_x$  se calcula así:

$$T_x = dh \quad (17)$$

Sustituyendo  $d$  en (14), se obtiene el valor del parámetro  $\alpha$ :

$$\alpha = \arctan \left( \frac{l'_t (1 - dl_{t-1}) - l_{t-1}}{1 - dl_{t-1} + l'_t l_{t-1}} \right) \quad (18)$$

Hasta aquí, la parte matemática del algoritmo no difiere en casi nada a la presentada en [2]. Sin embargo, en la práctica se ha comprobado que se obtienen resultados mucho más fiables si el valor de  $T_x$  se obtiene directamente del parámetro  $x_0$  del modelo clotoidal que se había empleado previamente para orientar la búsqueda de las líneas en la imagen. Este parámetro  $x_0$  indica el desplazamiento lateral del vehículo con respecto al centro del carril (Fig. 10). De esta forma, se calcula  $d$  sin tener que recurrir a una ecuación de segundo orden, evitando así tener que trabajar con raíces imaginarias, que no tienen sentido físico:

$$d = \frac{T_x}{h} \quad (19)$$

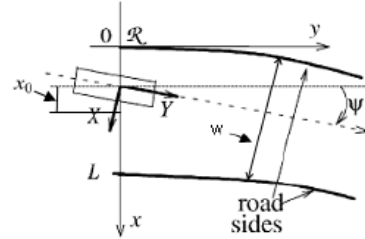


Fig. 10. Vista superior del modelo clotoidal de la carretera

Una vez obtenido  $d$  se calcula  $\alpha$  según (18). Sin embargo, se ha introducido la siguiente innovación, la cual permite obtener unos valores más robustos para el parámetro  $\alpha$ , ya que permite reducir en un 50% los errores en la estimación de esta variable. En lugar de utilizar para su cálculo la información de una de las dos líneas, se hace uso de la correspondiente a las dos líneas que delimitan el carril por el que circula el vehículo, de la siguiente manera:

$$\begin{aligned} \alpha_l &= \arctan \left( \frac{l'_t (1 - dl_{t-1}) - l_{t-1}}{1 - dl_{t-1} + l'_t l_{t-1}} \right) \\ \alpha_r &= \arctan \left( \frac{r'_t (1 - dr_{t-1}) - r_{t-1}}{1 - dr_{t-1} + r'_t r_{t-1}} \right) \end{aligned} \implies \alpha = \frac{\alpha_l + \alpha_r}{2} \quad (20)$$

Tras el cálculo de  $T_x$  y  $\alpha$ , es necesario compensar las líneas de la carretera por los efectos producidos por estos dos parámetros, puesto que las pendientes de las líneas compensadas serán usadas para estimar los parámetros en el siguiente

frame. Para compensar las líneas, en primer lugar se rota la cámara alrededor del eje  $Z_c$  por un ángulo  $-\alpha$ .

$$Rot(Z_c, -\alpha) = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (21)$$

Según esta transformación, las nuevas coordenadas  $(x'_c, y'_c, z'_c)$  de un punto con coordenadas en el sistema de la cámara  $(x_c, y_c, z_c)$  se calculan como sigue:

$$\begin{bmatrix} x'_c \\ y'_c \\ z'_c \end{bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \quad (22)$$

Sean  $(x_i, y_i)$  y  $(x'_i, y'_i)$  las coordenadas en la imagen resultado de proyectar los puntos  $(x_c, y_c, z_c)$  y  $(x'_c, y'_c, z'_c)$  en el plano imagen, respectivamente. De la expresión anterior se obtiene la relación entre  $(x_i, y_i)$  y  $(x'_i, y'_i)$ :

$$\begin{cases} x'_i = x_i \cos \alpha + y_i \sin \alpha \\ y'_i = -x_i \sin \alpha + y_i \cos \alpha \end{cases} \quad (23)$$

Después, hay que trasladar la cámara un valor  $T_x$  a lo largo del eje  $X'_c$  resultante de la rotación anterior. Las nuevas coordenadas  $(x''_i, y''_i)$  se calculan:

$$\begin{cases} x''_i = x'_i - \frac{T_x}{h} y'_i \\ y''_i = y'_i \end{cases} \quad (24)$$

Por lo tanto, dado un punto  $(x, y)$  perteneciente a una línea de la carretera, las expresiones (23) y (24) permiten compensar los efectos causados por  $T_x$  y  $\alpha$ . Si la transformación de dicho punto es  $(x'', y'')$ , la pendiente de la línea compensada, teniendo en cuenta que el punto de fuga sigue estando en el centro de la imagen, será:

$$s'' = \frac{0 - y''}{0 - x''} = \frac{y''}{x''} \quad (25)$$

## V. CLASIFICACIÓN DEL MOVIMIENTO Y SUAVIZADO

La tercera fase del algoritmo implementado consiste en clasificar el movimiento y suavizar los parámetros con objeto de conseguir la deseada estabilidad. El movimiento estimado de la cámara, caracterizado por los parámetros  $T_x$ ,  $\alpha$ ,  $\beta$  y  $\gamma$ , contiene tanto movimientos esperados como no esperados. Por ello es necesario clasificar los movimientos, discriminando entre unos y otros. Además, los movimientos esperados deben ser filtrados o suavizados, debido a que contienen ruido, imprecisión de cálculo y extracción imperfecta de las características globales del entorno.

En lugar de separar el proceso de clasificación del movimiento y el suavizado, se hacen en un mismo paso. Para ello, se utiliza un filtro de Kalman [3]. El filtro de Kalman determina el estado de un sistema en función de la predicción del estado a partir de sus valores anteriores y del estado derivado de la medida actual. En este cálculo intervienen las matrices de covarianza del sistema y de la medida. Cuanto mayor sea la diferencia entre el estado predicho y el derivado de la observación, el filtro tiende más hacia la predicción

puesto que la medida puede ser ruidosa o errónea. Por lo tanto, el filtro de Kalman realiza un suavizado secuencial de los estados.

Se utiliza un filtro de Kalman para cada uno de los cuatro parámetros  $T_x$ ,  $\alpha$ ,  $\beta$  y  $\gamma$ , dando como resultado los valores que se utilizan en la siguiente etapa.

## VI. COMPENSACIÓN DE LA IMAGEN

La última fase del algoritmo consiste en compensar la imagen de entrada para obtener la estabilizada de salida. Sólo los parámetros de rotación  $\alpha$ ,  $\beta$  y  $\gamma$  son utilizados, puesto que el efecto causado por el movimiento de la cámara en dirección perpendicular al movimiento del vehículo es mucho menor que el causado por éste último. Por tanto, en esta fase se ignora  $T_x$ .

Supóngase que la imagen de entrada se obtiene a partir de una cámara que está girada unos ángulos  $\alpha$ ,  $\beta$ ,  $\gamma$  alrededor de los ejes  $X_c$ ,  $Y_c$ ,  $Z_c$  respectivamente, donde  $\alpha$ ,  $\beta$  y  $\gamma$  son los valores estimados en la sección IV. La imagen estabilizada se obtendrá de la misma cámara, pero en vez de estar rotada  $\alpha$ ,  $\beta$  y  $\gamma$ , estará girada  $\alpha'$ ,  $\beta'$  y  $\gamma'$  respecto a los mismos ejes, donde  $\alpha'$ ,  $\beta'$  y  $\gamma'$  son los valores filtrados obtenidos en la sección V. Sea  $p$  un punto con coordenadas  $(x_c, y_c, z_c)$  y  $(x'_c, y'_c, z'_c)$  en el primer y en el segundo caso, respectivamente. La relación entre unas y otras coordenadas es la siguiente:

$$\begin{bmatrix} x'_c \\ y'_c \\ z'_c \end{bmatrix} = R_1 R_2 \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \quad (26)$$

donde:

$$\begin{aligned} R_1 &= Rot(Y'_c, -\gamma) \cdot Rot(X'_c, -\beta) \cdot Rot(Z_c, -\alpha) \\ R_2 &= Rot(Z'_c, \alpha') \cdot Rot(X'_c, \beta') \cdot Rot(Y_c, \gamma') \end{aligned} \quad (27)$$

$$\text{Sea } R = R_1 \cdot R_2 = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix} \text{ y sean } (x_i, y_i)$$

y  $(x'_i, y'_i)$  las coordenadas en la imagen resultado de la proyección de  $(x_c, y_c, z_c)$  y  $(x'_c, y'_c, z'_c)$  en el plano imagen. La relación entre unas y otras coordenadas se obtiene de (26) y de las ecuaciones del modelo pinhole [4]:

$$\begin{cases} x'_i = f \cdot \frac{x'_c}{z'_c} = f \cdot \frac{a_1 x_c + a_2 y_c + a_3 z_c}{c_1 x_c + c_2 y_c + c_3 z_c} = f \cdot \frac{a_1 x_i + a_2 y_i + f a_3}{c_1 x_i + c_2 y_i + f c_3} \\ y'_i = f \cdot \frac{y'_c}{z'_c} = f \cdot \frac{b_1 x_c + b_2 y_c + b_3 z_c}{c_1 x_c + c_2 y_c + c_3 z_c} = f \cdot \frac{b_1 x_i + b_2 y_i + f b_3}{c_1 x_i + c_2 y_i + f c_3} \end{cases} \quad (28)$$

En la práctica, la fase de compensación consiste en recorrer la imagen de salida, la imagen estabilizada, inicialmente con valores desconocidos, de forma que cada pixel  $(x'_i, y'_i)$  se hace corresponder con cada pixel  $(x_i, y_i)$  de la imagen de entrada no estable. De esta forma, el pixel en la imagen de salida se pone al mismo valor que el correspondiente pixel de la imagen de entrada. Para determinar la relación entre unas y otras coordenadas, se parte de la ecuación (26), pero aplicando la transformación inversa.

## VII. RESULTADOS EXPERIMENTALES

Se han llevado a cabo una serie de experimentos para comprobar la eficiencia de la técnica propuesta, utilizando tanto datos reales (imágenes adquiridas por una cámara embarcada en un vehículo) como artificiales, obtenidos a partir de los reales pero introduciendo una serie de perturbaciones a las imágenes. Aquí sólo se presentan los primeros, ya que son de un mayor interés. Los datos artificiales se emplearon en un principio para intuir cómo se iba a comportar el algoritmo. En la Fig. 11 puede verse la evolución de los parámetros  $\alpha$ ,  $\beta$  y  $\gamma$  en función del frame para un ejemplo concreto, tanto antes como después de aplicar el algoritmo de estabilización. En la Fig. 12 se muestra la variación del punto de fuga sobre el plano imagen antes y después de aplicar la estabilización, para el mismo ejemplo. Claramente la variación de los parámetros de la cámara después de estabilizar es mucho menor que antes de la estabilización, así como la variación del punto de fuga.

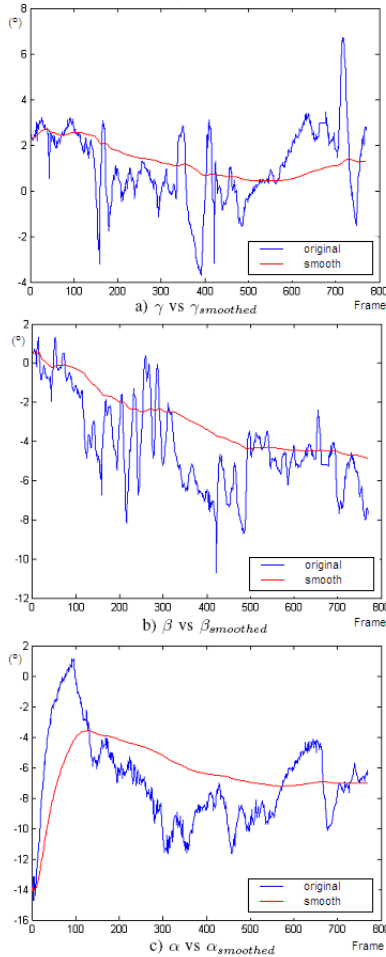


Fig. 11. Parámetros  $\alpha$ ,  $\beta$  y  $\gamma$  en función del frame

Según se ha comentado en la sección I, el propósito del estudio de esta técnica de estabilización es comprobar su eficacia en un sistema Light-Beam desarrollado por el Departamento de Electrónica de la Universidad de Alcalá, el cual emplea técnicas de visión artificial. En concreto, se había comprobado que uno de los principales problemas que plantea este sistema

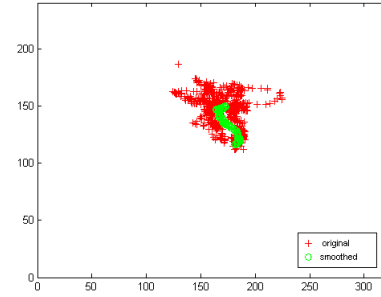


Fig. 12. Punto de fuga antes y después de estabilizar

es la estimación de distancias a la que se encuentran el resto de vehículos del entorno, puesto que este cálculo se realiza a partir de sus coordenadas  $(u, v)$  en la imagen. Se había demostrado que esta estimación de distancias era muy irregular debido a la inestabilidad de las imágenes. Sin embargo, se ha comprobado que la técnica de estabilización propuesta es capaz de enfrentarse a este problema, dando lugar a unas estimaciones de distancias muy robustas y regulares. En la Fig. 13 puede verse el cálculo de la distancia a la que se encuentra un vehículo que se aproxima en sentido contrario en función del frame, tanto antes como después de aplicar el algoritmo de estabilización propuesto. Puede comprobarse que, antes de estabilizar la imagen, la gráfica posee grandes oscilaciones, dando lugar a una estimación de distancia errónea. Mayor es el error cuanto mayor es la inestabilidad de la imagen. Sin embargo, una vez que se aplica el algoritmo de estabilización, se obtiene una función decreciente, como es lógico dado que se trata de la evolución de la distancia a la que se encuentra un vehículo que se encuentra aproximándose al nuestro.

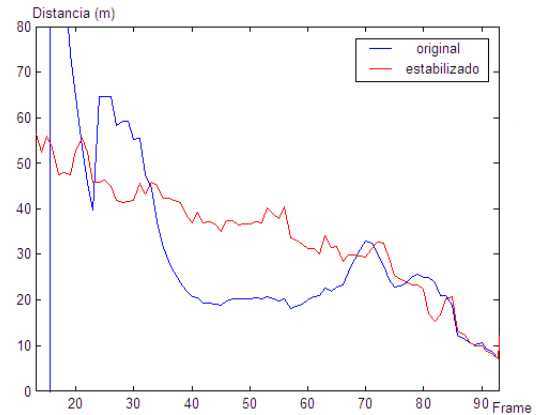


Fig. 13. Estimación de distancia para un vehículo que se aproxima en sentido contrario

## VIII. CONCLUSIONES

La técnica de estabilización propuesta, además de conseguir una buena estabilización de imágenes, dando lugar a una robusta estimación de distancias para mejorar la funcionalidad de un sistema Light-Beam, posee una serie de ventajas que hacen interesante su uso en otro tipo de sistemas que empleen técnicas de visión artificial. En primer lugar, permite calcular movimientos en el mundo 3D a partir de imágenes



2D. Además, lo hace a partir de unas pocas características presentes en la imagen muy fáciles de extraer y sobre las que ya existen muchas técnicas de detección fiables. Se trata únicamente de las líneas de la carretera y el punto de fuga, calculado como la intersección en el plano de las mismas. En tercer lugar, el algoritmo propuesto es válido tanto para ambientes nocturnos como diurnos. Y por último, permite el cálculo de los movimientos de la cámara a partir de la información extraída en el frame actual y en el anterior, sin ser necesario hacer un procesamiento *offline* de los datos. De esta forma, el algoritmo propuesto puede trabajar sobre aplicaciones en tiempo real, si bien sólo se ha estudiado con videos previamente grabados. Así pues, para el futuro queda la tarea de probar y optimizar el algoritmo para que funcione perfectamente *online*, pero la experiencia nos dice que este algoritmo puede ser empleado en aplicaciones en tiempo real.

Entre las desventajas que presenta la técnica propuesta es que no funciona en carreteras en las que las líneas que delimitan la misma no están pintadas. Si bien la mayoría de las carreteras actuales ya poseen marcas viales, lo cierto es que aún quedan muchas, especialmente secundarias y comarcales, que no son así. Es más, si las líneas se encuentran ocultas debido a retenciones de tráfico o están cubiertas por barro o nieve, el algoritmo no funcionaría, aunque una vez que las líneas apareciesen en la imagen, el sistema resurgiría tras unos pocos frames. Sin embargo, la flexibilidad del algoritmo propuesto hace que pueda ser combinado con cualquier otra técnica en la que la detección de las líneas de la carretera (o los bordes de la misma en el caso de no existir las líneas) sea más completa, de forma que este tipo de problemas se resuelvan.

Entre las características particulares del algoritmo, destaca la robustez a la hora de detectar las líneas de la carretera, siempre que éstas estén presentes y se encuentren en buenas condiciones, como ya se ha indicado. Esto es así gracias a la combinación que se ha realizado de técnicas de detección de líneas basadas en la transformada de Hough junto con la proyección sobre el plano imagen de un modelo clotoidal de la carretera. En esto consiste la primera fase del algoritmo. Para que la detección de las líneas sea más fiable, ha sido necesario aplicar un filtro de realce de bordes sobre la imagen de entrada. Es muy importante señalar que cuanto más fiable sea la detección de las líneas de la carretera, mejor será la estimación de los parámetros de movimiento de la cámara y mejor se comportará el algoritmo.

La segunda fase, en la que se estima el movimiento que ha sufrido la cámara, se reduce a calcular cuatro parámetros de movimiento: uno de traslación ( $T_x$ ) y tres de rotación ( $\alpha$ ,  $\beta$  y  $\gamma$ ), si bien el de traslación es despreciable a la hora de realizar la compensación de la imagen, aunque necesario para la estimación de los otros tres. Además, dos de los parámetros de rotación se calculan de una forma muy fácil, únicamente a partir de la posición del punto de fuga. Se ha comprobado experimentalmente que los errores que se producen a la hora de estimar estos parámetros,  $\beta$  y  $\gamma$ , son muy pequeños siempre que la estimación de la posición del punto de fuga sea buena, con un error no superior a  $\pm 4$  o  $\pm 5$  píxeles. Sin embargo, no se pueden extraer las mismas

conclusiones con el tercer parámetro en discordia:  $\alpha$ . Para su cálculo es necesario conocer cierta información tanto del frame actual como del previo. Esto requiere de un *buffer* en el que guardar la información correspondiente al frame anterior. Además, el mayor problema que plantea el parámetro  $\alpha$  es su gran dependencia con el parámetro  $T_x$ . Se ha comprobado empíricamente que un error de estimación cometido en  $T_x$  de  $\pm 10$  cm da lugar a un error en  $\alpha$  de  $\pm 5^\circ$ . Así pues, queda pendiente para el futuro conseguir una mejor estimación del parámetro  $T_x$ , pues si bien su valor se obtiene directamente de uno de los parámetros que definen el modelo clotoidal de la carretera, pueden utilizarse otras formas más fiables de extraer ese valor. Una posible opción puede consistir en aproximar los bordes de la carretera mediante curvas spline, en lugar de con un sencillo modelo trapezoidal como el que se ha utilizado.

Sin embargo, es sumamente importante señalar que, en la práctica, los movimientos que más afectan a la estabilidad de la imagen para un vehículo que se encuentra circulando por una carretera son aquellos que modifican el *pitch* de la cámara, relacionado con el parámetro  $\beta$  definido en este trabajo. Una plataforma sobre la cual vaya montada la cámara dentro del vehículo se ve mucho más afectada por inestabilidades en ese sentido que por movimientos en  $\alpha$  o en  $\gamma$ , de forma que no va a ser demasiado problemático el hecho de que no se consiga estimar suficientemente bien el valor del parámetro  $\alpha$ .

En resumen, el algoritmo propuesto ha demostrado ser muy efectivo para la aplicación sobre la que se había planeado su uso: un sistema Light-Beam o "Haz Inteligente", como puede verse en las gráficas de la sección VII, las cuales muestran la estimación de distancias a la que se encuentran vehículos que circulan en sentido contrario tanto antes como después de estabilizar.

#### AGRADECIMIENTOS

Este trabajo ha sido financiado mediante fondos del Ministerio de Educación y Ciencia a través de los proyectos MOVICON (TRA2005 - 08529 - 02) y CABINTEC (PSE - 370100 - 2007 - 2), así como de la Comunidad de Madrid a través del proyecto Robocity2030 (CAM - S - 0505 / DPI / 000176).

#### REFERENCES

- [1] P.F. Alcantarilla, L.M. Bergasa, P. Jiménez, M.A. Sotelo, I. Parra, D. Fernández, S.S. Mayoral, *Night Time Vehicle Detection for Driving Assistance LightBeam Controller*, IEEE Intelligent Vehicles Symposium, April 2008, Eindhoven, The Netherlands.
- [2] Y.-M. Liang, H.-R. Tyan, S.-L. Chang, H.-Y.M. Liao, S.-W. Chen, *Video stabilization for a camcorder mounted on a moving vehicle*, IEEE Transactions on Vehicular Technology, Vol: 53, No: 6, November 2004, pp: 1636-1648.
- [3] R. E. Kalman, *A new approach to linear filtering and prediction problems*, Transactions of the ASME - Journal of Basic Engineering, Vol: 82, pp: 35-45, 1960.
- [4] R. Hartley, A. Zisserman, *Multiple View Geometry in computer vision*, Cambridge University Press, 2001.
- [5] A. de la Escalera, *Visión por computador. Fundamentos y métodos*, Prentice-Hall, 2001.
- [6] E. D. Dickmanns, B. D. Mysliwetz, *Recursive 3-D road and relative ego-state recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol: 14, No: 2, February 1992, pp: 199-213.