

Jorge Cruz Pérez

1 de diciembre de 2003

Resumen

En este trabajo revisamos una técnica de detección y seguimiento de los ejes de un camino basados en transformaciones geométricas y en un rápido procesamiento morfológico, presentamos un análisis realizado a los trabajos de investigación de A. Brogui [1] y [2] y B. Poupard [3], además de ello una técnica basada en estéereoscopia para la de detección de obstáculos, esto implementado solo con los datos visuales adquiridos mediante camaras instaladas en vehículos en movimiento, presentando un sistema de *bajo costo* capaz de alcanzar un desempeño en tiempo real en la detección de caminos y robusto a condiciones impuestas por las condiciones externas.

1. Introducción

El tema de detectar carreteras, más en específico detectar los ejes del camino y la posición del vehículo sobre la misma ha sido desde varios años atras un tema de investigación interesante debido a la gran utilidad que esto tendría, de hecho se ha estimado que alrededor de un 30 % de los accidentes automovilísticos fatales ocurridos son atribuidos a la inatención del conductor y fatiga. Asípodríamos imaginar un sistema de control que nos permitiría crear un automóvil autónomo como un sistema final, completamente desarrollado con un sistema de detección de caminos que haya sido completamente probado contra errores. O imaginarnos que una de las primeras aplicaciones que tendría es advertir al conductor de si esta abandonando el camino al momento de sentirse cansado o somnoliento, esto con alguna alarma o algún método para advertir o despertar al conductor.

Otro ejemplo sería un sistema de control el cual permitiese construir un vehículo autónomo, mejorando la seguridad. Asi como también para mejorar el tráfico, o vehículos especialmente equipados que puedan viajar bajo control computarizado.

Además de esto el desarrollo de este tipo de sistemas requiere de un costo mínimo en la implementación, en el hardware, además de cambios mínimos en la infraestructura carretera.

Los principales problemas con los que nos encontramos en la detección de los limites de la carretera (lineas) son: 1) La presencia de sombras, produciendo

artefactos sobre la superficie del camino, y con esto alterando la textura, y 2) la presencia de otros vehículos sobre el camino, ocultandonos parcialmente el camino, es decir, nuestra búsqueda se basa en la localización de un patrón específico (líneas del camino), aunque también se han desarrollado sistemas que trabajen sobre caminos no estructurados (sin líneas sobre el camino) o sobre terreno sin pavimento. Basados en transformaciones geométricas y en un rápido procesamiento morfológico, y debido a las limitaciones para el análisis de ambientes estructurados (caminos con las líneas que lo delimitan pintados sobre la superficie) nos permite el uso de modelos simples del camino, así como un procesamiento monocular monocromático de imágenes.

Respecto a la localización de obstáculos sobre la carretera, las técnicas usadas para la detección de objetos están basadas en la búsqueda de patrones específicos, y algunas otras características como forma, simetría, etc., sin embargo, si tenemos un obstáculo, es decir, cualquier objeto que obstruye el camino, nuestra búsqueda consiste entonces en detectar *el espacio libre*, en lugar de algún patrón específico. Pero esto se pueden usar diferentes técnicas tales como: 1) análisis del campo de flujo óptico, o 2) el procesamiento de imágenes estéreo; ambas técnicas requieren de dos o más imágenes incrementando la complejidad computacional.

La ventaja del análisis de imágenes estéreo en vez de una secuencia de imágenes monoculares es la posibilidad de directamente detectar la presencia de objetos, la cual, en el caso de análisis de flujo óptico, es derivada indirectamente del análisis del campo de velocidad, además de otras limitaciones para velocidades cero.

Debido al campo específico de aplicación, debemos estar basados en una arquitectura de cómputo estándar en donde su construcción involucre el menor costo posible ofreciendo ventajas de cómputo adecuadas.

En este trabajo está basado principalmente por los métodos propuestos por:

- Massimo Bertozzi and Alberto Broggi, [1].
- Alberto Broggi, [2] que es un trabajo preliminar de donde se desprende el método anterior [1], y
- Baptiste Poupard and M. Stephane Mallat, basado en la teoría de filtros [3].

El siguiente trabajo está organizado de la manera siguiente, en la Secc. 2 describimos los métodos propuestos y hacemos el análisis correspondiente, describiendo los pasos del algoritmo de implementación. En la Secc. 3 revisamos los resultados obtenidos en cada artículo, presentamos las ideas generales de cómo se lleva a cabo la implementación y los resultados de los métodos propuestos. Finalmente en la Secc. 4 presentamos las conclusiones al respecto.

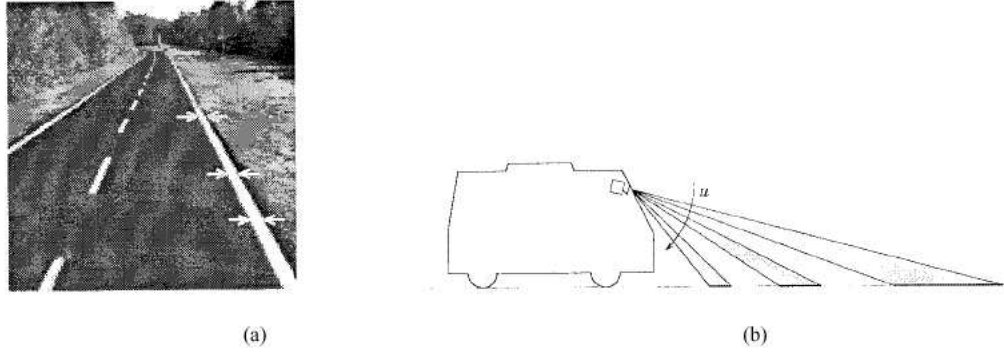


Figura 1: a) El ancho de las Marcas del camino varían de acuerdo a su posición en la imagen, b) Debido al efecto de perspectiva, diferentes pixeles representan diferentes porciones del camino.

2. Detección de Líneas y Caminos

2.1. El problema a resolver: *Seguimiento del Camino*

El problema a resolver es el siguiente: detectar el camino y la posición del vehículo sobre este. En general se describe un método para encontrar las marcas del camino en un espacio de trabajo adecuado, la forma de reconocer un obstáculo a distancia y la integración de todo esto en un sistema global montado en vehículos de prueba. (vehículo MOB-LAB y vehículo autónomo ARGO [2]).

2.1.1. Las observaciones

Las observaciones que tenemos son los diferentes cuadros (frames) de la secuencia, donde no tenemos solo el camino, sino también otros vehículos, panorama, casas o el cielo mismo. Hemos escogido transformar la imagen en una imagen binaria donde primero removemos el efecto de perspectiva para obtener una imagen vista desde arriba y después detectar las líneas de acuerdo a un patrón específico en el cambio de intensidades.

2.2. El Mapeo de Perspectiva Inverso

Debido al efecto de perspectiva por las condiciones de adquisición, el ancho de las líneas del camino van cambiando de acuerdo a la distancia desde la cámara. De hecho este efecto asocia diferentes significados a los diferentes pixeles de la imagen, dependiendo de su posición en la misma. Una vez removido este efecto de perspectiva, cada pixel representa la misma porción del camino, permitiendo una distribución homogénea de la cantidad de información (fig.1). Para remover este efecto es necesario conocer las condiciones específicas de adquisición (posición, orientación, óptica, ...) y la escena representada por

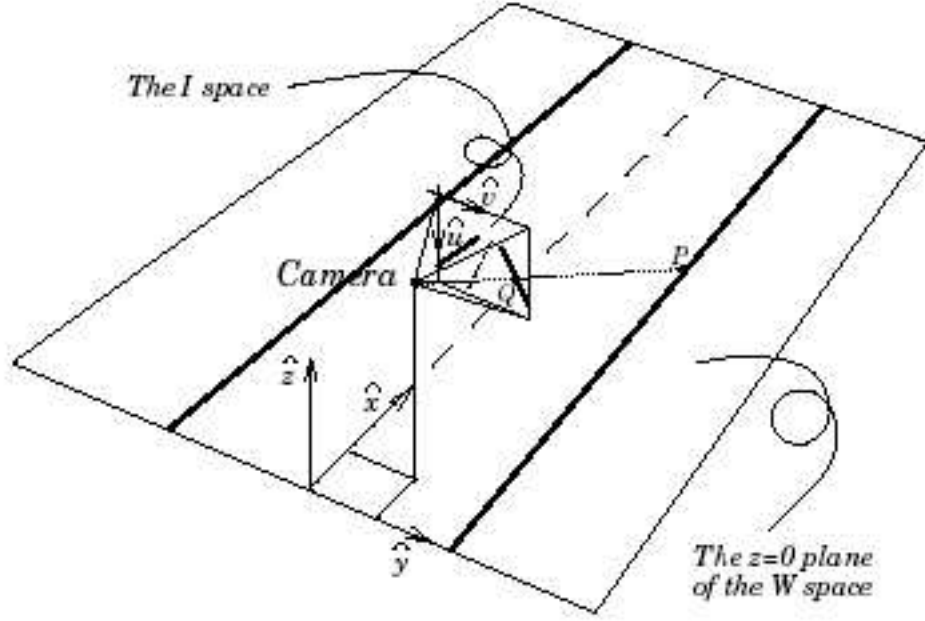


Figura 2: Relación entre los dos sistemas de coordenadas

la imagen (el camino, el cual suponemos plano), todo esto representa nuestro conocimiento *a priori*.

El procesamiento puede dividirse en dos pasos: primero, tomando en cuenta nuestro conocimiento *a priori*, como una transformación (remuestreo no uniforme), que nos genera una imagen en un nuevo dominio, donde detectar las características de interés se simplifica; segundo, realizar un procesamiento morfológico de bajo nivel.

El MPI (Mapeo de Perspectiva Inverso) nos permite transformar la imagen en perspectiva (la imagen sobre una pantalla) a una vista desde arriba. El siguiente procedimiento nos auxilia a remover este efecto, remapeando cada pixel hacia una diferente posición, teniendo así un nuevo arreglo de pixeles. Esta imagen resultado representa una perspectiva a una cierta altura desde arriba de la región del camino enfrente del vehículo.

Definamos dos Espacios Euclidianos de la siguiente manera:

- $\mathcal{W} = \{(x, y, z)\} \in E^3$ representa el espacio $3D$, donde definimos el espacio del mundo real,
- $\mathcal{I} = \{(u, v)\} \in E^2$ representa el espacio $2D$ de la imagen (coordenadas de la pantalla), donde la escena $3D$ es proyectada.

La imagen que tenemos de la cámara pertenece al espacio $\mathcal{I}$, mientras que la imagen transformada esta definida en $z=\theta$ del espacio $\mathcal{W}$. La fig.2 muestra la relación entre ambos espacios.

2.2.1. Mapeo del espacio $\mathcal{W}$ al espacio $\mathcal{I}$.

Para realizar esta transformación se deben conocer los siguientes parámetros:

- *punto de perspectiva*: es la posición de la cámara en $C = (l, h, d) \in \mathcal{W}$;
- *dirección de vista*: es el eje óptico $\hat{o}$ determinado por los siguientes ángulos:
 - ▷ $\tilde{\gamma}$: el ángulo formado por la proyección del eje óptico $\hat{o}$ sobre el plano $z=\theta$ y el eje x (como se muestra en la fig.3.(a)).
 - ▷ $\tilde{\theta}$: el ángulo formado por el eje óptico $\hat{o}$ y la horizontal (mostrado en la fig.3.(b)).
- *apertura*: la apertura angular de la cámara es 2α ;
- *resolución*: la resolución de la cámara $n \times n$.

Mapeo $\mathcal{I} \rightarrow \mathcal{W}$

Después de ciertas manipulaciones algebraicas y trigonométricas, el mapeo final $f : \mathcal{I} \rightarrow \mathcal{W}$ es una función de u y v dada por:

$$\begin{cases} x(u, v) = \frac{h}{\tan[(\tilde{\theta}-\alpha)+u\frac{2\alpha}{n-1}]} \times \cos[(\tilde{\gamma}-\alpha) + v\frac{2\alpha}{n-1}] + l \\ y(u, v) = \frac{h}{\tan[(\tilde{\theta}-\alpha)+u\frac{2\alpha}{n-1}]} \times \text{sen}[(\tilde{\gamma}-\alpha) + v\frac{2\alpha}{n-1}] + d \\ z = 0 \end{cases}$$

con $u, v = 0, 1, \dots, n-1$.

Mapeo $\mathcal{W} \rightarrow \mathcal{I}$

La transformación inversa $g : \mathcal{I} \rightarrow \mathcal{W}$ esta dada por:

$$\begin{cases} u(x, y) = \frac{\gamma(x, y) - (\tilde{\gamma} - \alpha)}{\frac{2\alpha}{n-1}} \\ v(x, y) = \frac{\theta(x, y) - (\tilde{\theta} - \alpha)}{\frac{2\alpha}{n-1}} \end{cases} \quad (1)$$

Con γ y θ dadas como:

$$\begin{cases} \gamma = \arctan\left(\frac{x}{y}\right) \\ \theta = \arctan\left(\frac{h}{\sqrt{x^2 + y^2}}\right) \end{cases} \quad (2)$$

Asi removemos el efecto de perspectiva y obtenemos una vista del plano $z = \theta$ en el espacio $\mathcal{W}$. Esto se logra recorriendo el arreglo de pixeles de coordenadas

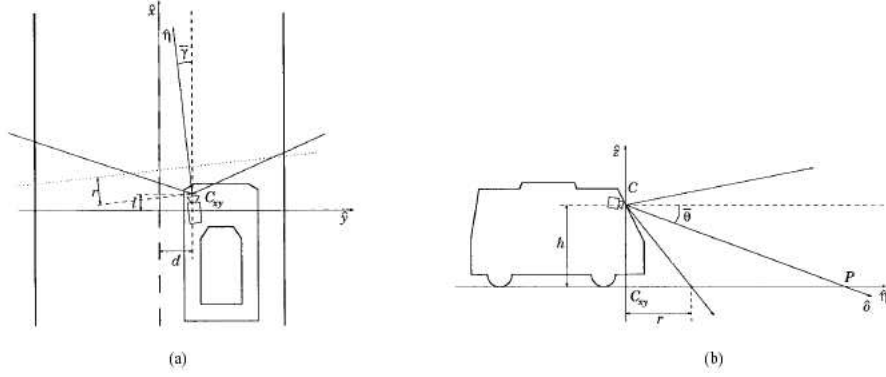


Figura 3: Planos xy y zn en el espacio $\mathcal{W}$

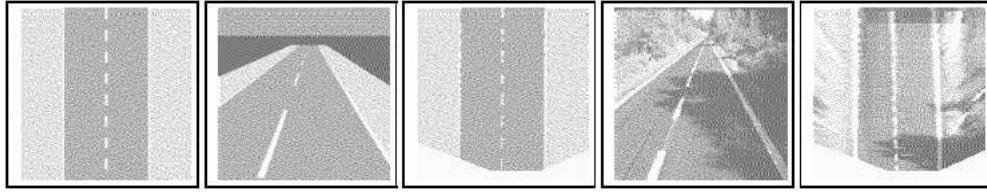


Figura 4: a) La imagen modelo, b) la proyección al espacio $\mathcal{I}$, c) el resultado de la reorganización de píxeles aplicado a b), d) la imagen original, e) la imagen mapeada.

$(x, y, 0) \in \mathcal{W}$ que forma la imagen remapeada para asociar cada uno de estos con su correspondiente punto de coordenadas $(u(x, y, 0), v(x, y, 0)) \in \mathcal{I}$.

La fig.4.a muestra la aplicación de este proceso aplicando (2) a la fig.4.b. Podemos notar, como se muestra en la fig.4.c que la porción de la parte inferior de la imagen queda indefinida, esto debido a la posición específica de la cámara con respecto al eje z , y la apertura angular de la misma.

Como ejemplo vease la fig.4.e que muestra el efecto de esta reorganización aplicado a una imagen real (fig.4.d), también se observa que el ancho de las marcas del camino (líneas) son invariantes dentro de toda la imagen.

2.2.2. Calibración de la cámara

Como hemos visto la importancia del sistema de adquisición de imágenes, la calibración del sistema de visión juega un papel muy importante. Los parámetros de calibración se pueden dividir en dos categorías.

- *parámetros intrínsecos* (apertura angular de la cámara 2α y resolución $n \times n$), los cuales son fijos.

- *parámetros extrínsecos* (punto y dirección de vision), los cuales pueden determinarse con mediciones y ajustarse.

2.3. Procesamiento de Bajo Nivel

2.3.1. Identificando las lineas del camino usando filtros basados en derivadas

Para esto podemos usar un algoritmo de detección de bordes basado en un umbral de la magnitud del gradiente, por ejemplo operadores de Roberts, Prewitt, usando máscaras de diferentes tamaños, reduciendonos los errores debido al efecto de ruido promediando con los valores vecinos.

Tambien podemos usar un filtro usando diferencias de segundo orden, localizando los cruces por cero, para dos dimensiones calculamos el Laplaciano, o usamos una máscara de convolución como por ejemplo:

$$\nabla^2 f(i, j) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \otimes f(i, j)$$

Interpretación en el espacio de frecuencias

Como vemos, estamos tratando de detectar bordes de baja a alta a baja intensidad, así nuestro algoritmo detecta varias áreas donde se presentan estos cambios, de hecho, en el espacio de *Fourier* aplicamos un filtro de pasa alta a la imagen. Desafortunadamente, el ruido está presente en estas altas frecuencias, de hecho también las diferencias de segundo orden son un filtro de pasa alta, que nos aumenta el ruido, así lo primero a realizar es aplicar un filtro de pasa baja para después convolucionar con la máscara escogida.

Filtros Gaussianos

Una buena manera de suavizar una imagen es convolucionarla con una Gaussiana, la desviación estándar es el único parámetro de nuestro filtro. La localización de los cruces por cero en la imagen resultado corresponde a la posición de los ejes. Con esto logramos una buena localización de los ejes e toda la imagen, este es el método utilizado en [3], sin embargo existen mejores condiciones como veremos a continuación.

2.3.2. Identificando las lineas del camino basados en los cambios de intensidades

Una vez hecha nuestra transformación de la imagen en $z = 0$ en el espacio $\mathcal{W}$, las líneas del camino están representadas por una línea brillante casi-vertical rodeada por una región oscura. Así los píxeles correspondientes a las líneas del camino tienen un valor mayor de brillo que las regiones a la derecha e izquierda de esta a una cierta distancia. Así nos enfocamos a determinar las transiciones horizontales de negro-blanco-negro y posteriormente un procesamiento de medio nivel para extraer la información y reconstruir la geometría del camino.

Extracción de las características de paralelismo: Por cada línea de la imagen $x = 0, 1, \dots, n - 1$, en cada píxel $P = (x, y)$ se compara su valor de intensidad $b(x, y)$ con sus vecinos a la izquierda y derecha a una cierta distancia $m : b(x, y - m)$ y $b(x, y + m)$ para $m \geq 1$.

La nueva imagen con valores $r(x, y)$ detecta la presencia de líneas de acuerdo a la siguiente expresión:

$$r(x, y) = \begin{cases} d_{+m}(x, y) + d_{-m}(x, y) & \text{si } d_{+m}(x, y) > 0 \text{ y } d_{-m}(x, y) > 0 \\ 0, & \text{de otra manera} \end{cases} \quad (3)$$

donde:

$$d_{+m}(x, y) = b(x, y) - b(x, y + m) \quad (4)$$

$$d_{-m}(x, y) = b(x, y) - b(x, y - m) \quad (5)$$

La elección de m depende del ancho de las líneas del camino, del proceso de adquisición y de los parámetros usados en IPM.

Debido a las diferentes condiciones del camino, las líneas pueden tener diferentes intensidades de brillo, pero manteniendo su superioridad respecto a sus vecinos horizontales. Aunque un valor de umbral adecuado nos da buenos resultados, podemos mejorar la imagen valiéndonos de la correlación vertical, es decir, llevamos a cabo una binarización adaptable.

Esta mejora se lleva a cabo mediante algunas iteraciones de *dilatación geodésica morfológica* para una estructura de 4 vecinos, donde:

$$c(x, y) = \begin{cases} 1, & \text{si } r(x, y) \neq 0 \\ 0, & \text{de otra manera} \end{cases} \quad (6)$$

es la *imagen de control*. El resultado de la dilatación geodésica es el producto de la imagen de control y el máximo valor calculado a lo largo de todos los píxeles pertenecientes a la estructura de 4 vecinos. La aplicación iterativa de esta dilatación corresponde al ensanchamiento de la vecindad, excepto en donde los valores de control valen cero. Haciendo esto tenemos nuestra imagen mejorada $e(x, y)$.

Más aún, si:

$$r(x, y) \neq 0 \implies \begin{cases} r(x, y - m) = 0 \\ r(x, y + m) = 0, \end{cases} \quad (7)$$

los píxeles de la imagen filtrada a una distancia m de una línea tienen un valor cero, de acuerdo a la ec.(6) su valor de control es cero, formando una barrera. Finalmente una binarización se lleva a cabo mediante un umbral adaptable:

$$t(x, y) = \begin{cases} 1, & \text{si } e(x, y) \geq \frac{m(x, y)}{k} \\ 0, & \text{de otra manera} \end{cases} \quad (8)$$

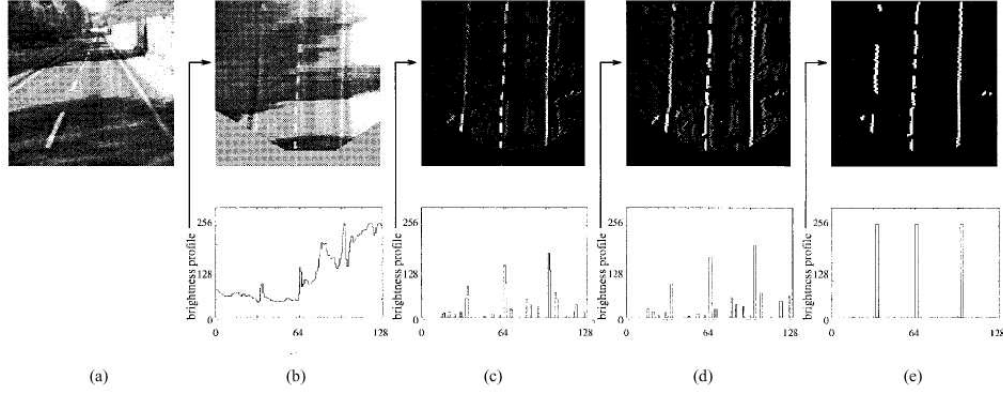


Figura 5: Secuencia de imágenes en el procesamiento de bajo nivel. a) La imagen original, b) la imagen remapeada, c) Imagen filtrada, d) Imagen mejorada, e) Imagen binarizada

donde $e(x,y)$ representa la imagen mejorada, $m(x,y)$ es el máximo valor calculado dado una vecindad $c \times c$ y k es una constante.

Identificación de las características del camino: La meta en el procesamiento de medio-nivel es la determinación de la geometría del camino, comenzando con la imagen binarizada. Esta imagen es escaneada fila a fila y por cada línea i todos los pixeles no cero son considerados en pares, un pixel no cero puede representar el eje izquierdo, derecho o el eje central del camino, la fig.6 muestra las tres diferentes configuraciones del camino que un par de pixeles no cero puede representar. Cada una de estas posibles configuraciones produce un par (c_i, w_i) , donde c_i representa la coordenada del eje medio del camino y w_i representa su correspondiente ancho de carril. De cualquier forma, no todos los pares (c_i, w_i) corresponde a configuraciones válidas del camino, solo aquellas que satisfacen las siguientes condiciones son consideradas: $0 \leq c_i \leq N$; $w_i \leq (N/3)$; $c_i - w_i \leq (\frac{3}{4}N)$; $c_i + w_i \geq (N/4)$; donde N es el tamaño horizontal de la imagen.

Construimos un histograma con w_i , por cada línea de la imagen, el pico del histograma corresponde al valor mas frecuente del ancho de carril, para permitir una geometría no fija y también manejar las curvas filtramos el histograma con un filtro de pasa-baja, finalmente calculamos el valor W máximo.

Así, todos los pares (c_i, w_i) con $W - (W/4) < w_i < W + (W/4)$ son considerados; la imagen es recorrida línea a línea desde abajo, donde es más probable detectar en centro del camino, hacia arriba, construyendo una larga cadena de centros de carretera. Los valores w_i correspondiente a los centros c_i sobre la cadena son usados para reconstruir la geometría del camino.

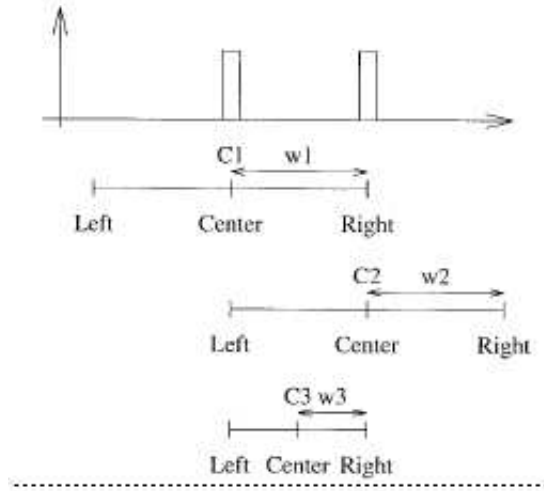


Figura 6: Tres configuraciones para los píxeles no cero

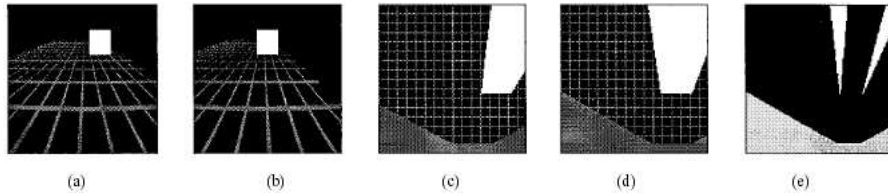


Figura 7: Objeto ideal, a) Vista izquierda, b) Vista derecha, c) Imagen remapeada izquierda, d) Imagen remapeada derecha, e) Diferencia entre c y d.

2.3.3. Detección de obstáculos

Usando el algoritmo de IPM con los parámetros apropiados para imágenes estéreo, diferentes mapeos de la imagen pueden ser obtenidos. Más aún el conocimiento de los parámetros del sistema de vision nos permite obtener los dos mapeos en correspondencia. Esto significa que, bajo la hipótesis de camino plano, pares de píxeles que tienen la mismas coordenadas en las dos imágenes remapeadas son puntos homologos y representan los mismos puntos en el plano del camino.

La hipótesis de un camino plano puede ser verificada calculando la diferencia entre las dos imágenes remapeadas: un obstáculo (cualquier cosa obstruyendo enfrente del camino) es detectada si la imagen diferencia presenta grandes clusters de píxeles no cero que tienen una forma específica. Debido a la diferen-

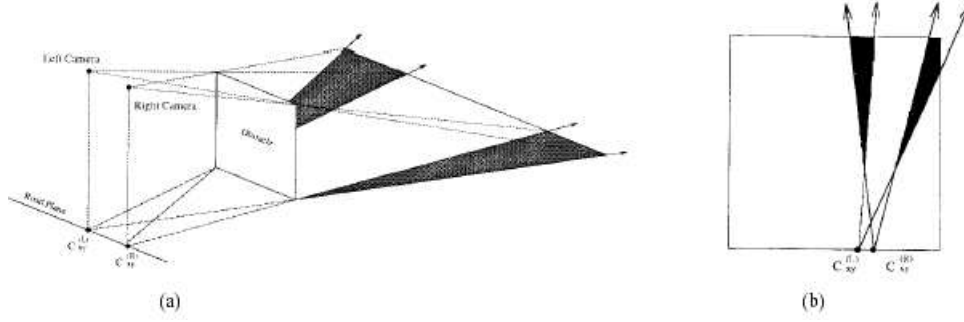


Figura 8: Relación entre obstáculos y la posición de las cámaras, a) Vista en Perspectiva, b) Vista desde el plano del camino.

cia de posición de las dos cámaras, la imagen diferencia solo puede ser calculada sobreponiendo el área de las dos imágenes mapeadas del camino.

Ahora, por ejemplo, un obstáculo que es un cuadrado regular, la imagen diferencia es transformada en dos triángulos. Así la detección de obstáculos está basada en la localización de estos pares de triángulos, que se derivan de una medición cuantitativa de la forma y posición de dos triángulos correspondientes (fig.7 y 8). Desafortunadamente, debido a la forma irregular, textura, o a la no homogeneidad del brillo de los obstáculos reales, esta localización de triángulos se vuelve difícil. Sin embargo, en la imagen diferencia existen algunos clusters de píxeles con una forma cuasi-triangular que pueden ser reconocidos.

1. *Histograma Polar*: Un *histograma polar* es usado para la detección de triángulos: dado un punto F en el plano $z = 0$ del espacio $\mathcal{W}$ (llamado *foco*), el histograma polar se calcula recorriendo la imagen diferencia y contando el número de píxeles arriba de cierto umbral por cada línea recta con origen en el foco (fig 9).

Cuando F se extiende sobre la prolongación del eje de un triángulo, el histograma polar presenta un discontinuidad, así la localización de un triángulo es hecha de acuerdo a la siguiente propiedad: la prolongación de los ejes de un triángulo intersecta ambas proyecciones $C_{xy}^{(L)}$ y $C_{xy}^{(R)}$ de las dos cámaras sobre el plano del camino, ver fig 8. Debido a la pequeña distancia entre $C_{xy}^{(L)}$ y $C_{xy}^{(R)}$, en vez de calcular dos diferentes histogramas consideramos solo uno. El foco de este histograma se localiza al centro de $C_{xy}^{(L)}$ y $C_{xy}^{(R)}$; en este caso el histograma presenta un claro pico por cada triángulo. Estos picos pueden tener diferentes características, tales como amplitud o ancho, dependiendo de la distancia del obstáculo, al ángulo de visión y la diferencia de intensidad entre el fondo y el obstáculo por sí mismo.

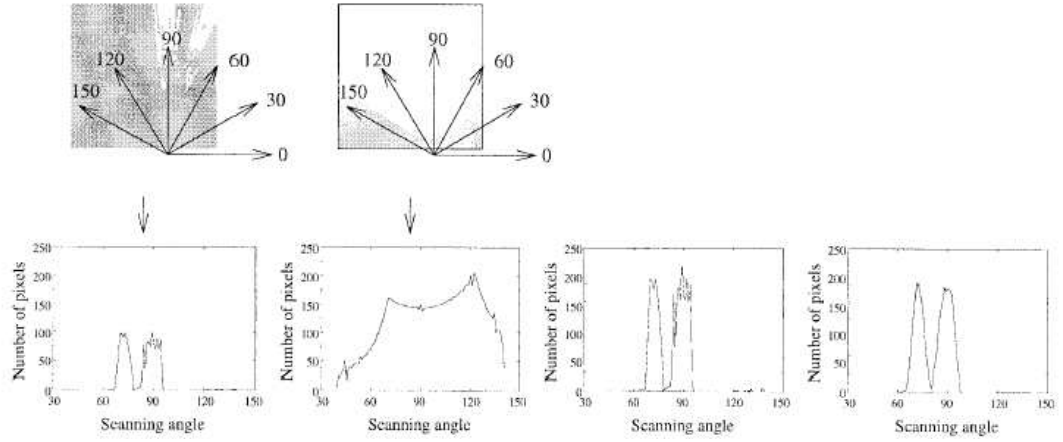


Figura 9: Pasos para el cálculo del histograma polar

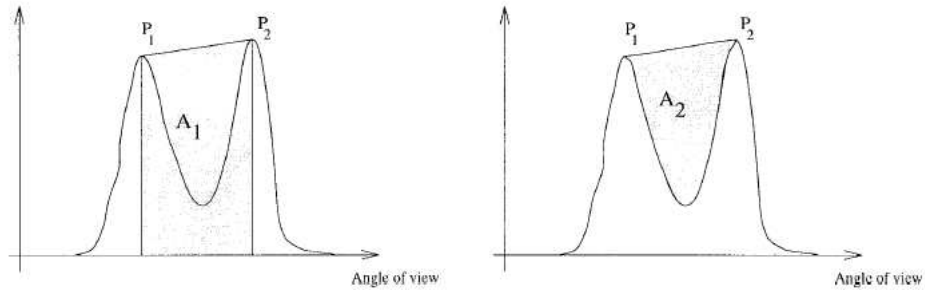


Figura 10: Si la relación entre las áreas A_1 y A_2 es mayor que un umbral, los picos se unen, lo cual corresponde a un mismo obstáculo

2. *Unión de Picos:* Dos o mas picos pueden ser unidos de acuerdo a diferentes criterios, tales como amplitud similar, proximidad o agudeza. De acuerdo al análisis de varios estudios nos permiten determinar un parámetro que involucra todas estas cantidades, así definimos R la relación entre las áreas A_1 y A_2 . Si R es mayor que un cierto umbral, dos picos adyacentes se consideran generados por un mismo obstáculo y entonces se unen (figura 10).
3. *Estimación de la distancia al obstáculo:* La imagen diferencia también puede usarse para estimar la distancia al obstáculo. Por cada pico en el histograma, calculamos un histograma radial recorriendo un sector específico de la imagen diferencia; el ancho α_i del sector es el ancho correspondiente al 80% de la amplitud del máximo pico en el histograma polar como se muestra en la figura 11. El número de píxeles arriba de un cierto umbral es calculado y el resultado normalizado. El histograma radial es analizado

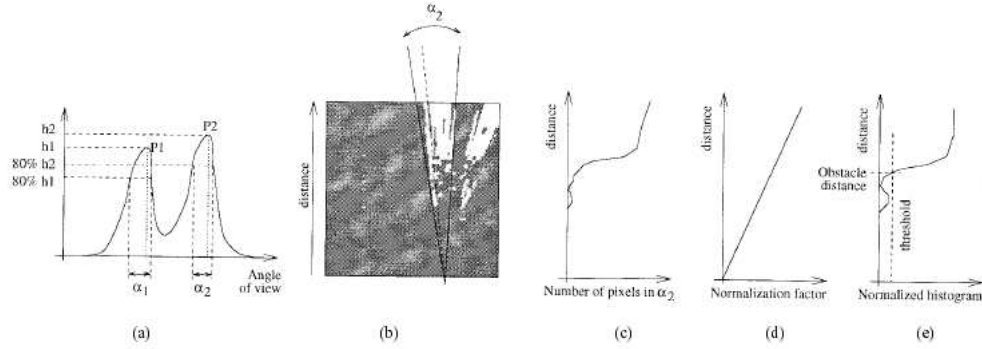


Figura 11: Pasos para el cálculo del histograma radial para el pico P_2 . a) Histograma Polar, b) Imagen diferencia binaria, c) Histograma Radial, d) Factor de Normalización, e) Histograma Radial Normalizado.

para encontrar esquinas de triángulos, los cuales representan los puntos de contacto entre obstáculos y el plano del camino, esto permite determinar la distancia mediante un umbral simple. (ver fig.11).

Los resultados se muestran con marcas negras sobre la imagen izquierda, conociendo su distancia y ancho del obstáculo.

3. Implementación, Desempeño y Propuestas

Debido al campo específico de aplicación, el tiempo de respuesta es un punto muy importante, la elección de la arquitectura de cómputo es entonces clave el diseño del sistema. En general debemos cuidar las características que permitan de integración y uso eficiente como son: 1) bajo costo, 2) bajo costo de operación, y 3) tamaño pequeño.

Para [3] los resultados en cuanto tiempo fueron:

dimensiones de la imagen	tiempo para calcular el IPM
100 x 100	0.04s
200 x 200	0.13s
400 x 400	0.49s

Los principales problemas con los que se enfrentan son:

- *En el Mapeo de Perspectiva Inverso*: se desconocen los valores exactos de los parámetros de la cámara, la manera en que los encuentran es empírica. Cuando se enfrentan a caminos no planos el algoritmo no trabaja bien, de hecho si la pendiente del camino rebasa los 5° empieza a fallar.

- *Detección del camino:* En general su algoritmo aun no trabaja en tiempo real, se desarrollo en C y estan optimizandolo, la tabla 2 muestra sus resultados.

dimensiones de la imagen binaria	tiempo para calcular los parámetros
100 x 100	2s
200 x 200	7s
400 x 400	18s

Tabla 2.

EN la implementación de A. Broggi [1] y [1], la arquitectura de cómputo que utilizan esta diseñada específicamente. Se analiza el tiempo de cálculo, procesamiento paralelo, y utilizan el Procesador Paralelo para Prueba y Análisis de Imágenes (PAPRICA [2]). Las características integradas al sistema PAPRICA para uso eficiente en la implementación son: 1) Procesamiento Morfológico, 2) Soporte de hardware para tareas de vision piramidal, 3) Integración directa de un dispositivo de adquisición de imágenes, 4) Soporte de hardware para un rápido remapeo de imágenes. Además de esto algunas tareas de procesamiento de medio-nivel son ejecutados en una computadora huesped.

Análisis de desempeño:

1. *Adquisición de Datos:* Se adquieren un par de imágenes estéreo de 512x512 pixeles simultáneamente, y se escriben directamente en la memoria de PAPRICA. Al mismo tiempo, los resultados de cálculos anteriores se muestran en un monitor abordo como información util.
2. *Remapeo:* Las imágenes adquiridas son remapeadas en dos imágenes de 128x128 pixeles.
3. *Preprocesamiento para la Detección de Obstáculos:* La Diferencia entre las dos imágenes se calcula y se umbraliza, se aplica un filtro morfológico para remover el ruido. El resultado es una imagen binaria de 128x128 pixeles. Esta fase toma 25 ms.
4. *Preprocesamiento para la Detección de Lineas:* El resultado es una imagen binaria de 128x128 pixeles, toma alrededor de 34 ms.
5. *Detección de Obstáculos:* el resultado depende de los datos, el estimado del tiempo que requiere la computadora huesped es de 20-30 ms. Después se trasfiere los resultados a la memoria de PAPRICA.
6. *Detección de Lineas:* El cálculo también depende de los datos, el tiempo estimado es de 30 ms.
7. *Advertencias:* La última fase del cálculo es mostrar los resultados, y advertencias, si es necesario, al conductor.

El problema principal con el IPM es que, al igual que en el caso anterior, para pendientes elevadas del camino el algoritmo falla, produciendo una imagen

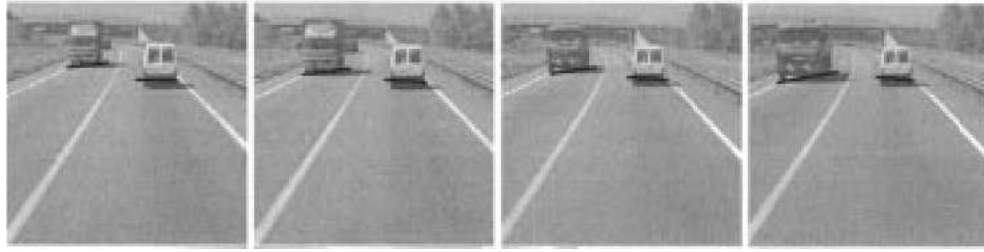


Figura 12: Algunos resultados en la detección de obstáculos, las marcas oscuras determinan la posición

irregular, en el caso de que una de las marcas del camino no sea visible produce una imagen mapeada incompleta que no resulta útil.

En el caso de detección de obstáculos, si este se encuentra bastante lejos (45-50 m.) no se logra identificarlo bien. También en los casos de obstáculos parcialmente visibles, o intensidades similares a las del camino.

4. Conclusiones

- a. Se presentaron un conjunto de herramientas para la detección de los ejes de carreteras y obstáculos, las dificultades envueltas, etc., donde el problema a resolver es aún un problema abierto a la investigación y desarrollo de nuevas técnicas, así como la mejora de las existentes, lo cual es en beneficio de la seguridad de los conductores, por lo que se debe lograr cubrir una amplia gama de situaciones y donde el sistema desarrollado sea robusto a todas estas condiciones impuestas.
- b. Una gran debilidad del método propuesto en la detección de las líneas, es la condición de marcas (líneas) estén pintadas sobre el pavimento, lo cual en lugares donde no existen carreteras pavimentadas o el mantenimiento no se realiza no se tienen las condiciones necesarias para su uso. Sin embargo, se ha demostrado que el método propuesto es robusto a condiciones de sombra o cambios de iluminación, cuando contamos con las condiciones físicas que exige el método.
- c. De acuerdo a los artículos que lei, y muchos más que fui encontrando posteriormente, existen muchos enfoques con los que se puede implementar un sistema de este tipo, resolviendo cada problema en particular de diferente manera, como tal es un área de investigación sumamente interesante y con aplicaciones a la vida real muy importantes.
- d. Un parte muy interesante es la implementación de un sistema similar con herramientas simples de cómputo, es decir, hacer una prueba física con una laptop como arquitectura de cómputo y cámaras web, montadas fijamente sobre algún vehículo y no solo esto sino tratar de mejorar los resultados

logrados. Es de hecho un objetivo que me he propuesto y ver que se puede lograr.

Referencias

- [1] Alberto Broggi, *Robust Real-Time Lane and Road Detection in Critical Shadow Conditions*. Dipartimento di Ingegneria dell'Informazione. Università di Parma. Proc. IEEE Int. Symp. Comput. Vis., Coral Gables, FL. Nov 19-21, 1995.
- [2] Alberto Broggi and Massimo Bertozzi, *GOLD, A Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection*. Dipartimento di Ingegneria dell'Informazione. Università di Parma. publicado en: IEEE Transactions on Image Processing, Vol. 7, No.1, January 1998.
- [3] Baptiste Poupard, and M. S. Mallat, *Road Edge Tracking Using Filtering Theory*, Image et Signal, Departement de Mathematiques Appliquees. Princeton University. 2000. <http://www.multimania.com/roadtracking>
- [4] M. Rossi, M. Aste, The IRST Driver's Assistance System. Istituto per la Ricerca Scientifica e Tecnologica, Trento, Italy, IRST-Technical Report # 9611-01.
- [5] A. Broggi, S. Berté, Vision-Based Road Detection in Automotive Systems: A Real-Time Expectation-Driven Approach. Dipartimento di Ingegneria dell'Informazione. Università di Parma. Journal of Artificial Intelligence Research, 3 (1995) 325-348.