



INAOE

Conjunto de componentes para el desarrollo de un instrumento de alerta específico hacia conductores de vehículos dedicado a la prevención de accidentes sobre objetos transversales.

por

Ing. Moisés Emilio Morales Velázquez

Tesis sometida como requisito
parcial para obtener el grado de:

**MAESTRO EN CIENCIAS EN LA ESPECIALIDAD DE
ELECTRÓNICA**

en el

Instituto Nacional de Astrofísica, Óptica y Electrónica

Noviembre 2011

Tonantzintla, Puebla

Director de tesis:

Dr. Jorge Francisco Martínez Carballido, INAOE

©INAOE 2011

Derechos Reservados

El autor otorga al INAOE el permiso de reproducir y distribuir copias de esta tesis en su totalidad o en partes.



RESUMEN

Hoy en día la cantidad de vehículos en un área urbana se incrementa rápidamente, donde el aumento de vehículos y personas en las calles de las ciudades desafortunadamente implica el aumento de los accidentes de tránsito, en México más de 17 mil personas mueren al año en este tipo de accidentes y más de la tercera parte son peatones [1] La distracción del conductor es la principal causa de estos accidentes.

Esta tesis presenta el desarrollo de un algoritmo para la detección de una persona que cruce transversalmente al vehículo, específicamente el zapato de la persona que apoya en el pavimento, esto provee una advertencia de tres niveles a un sistema avanzado de ayuda al conductor ante situaciones de riesgo de que un peatón esté presente en el camino del vehículo. La detección se realiza mediante la búsqueda de bordes característicos de un zapato con umbral adaptable para detectar una amplia variedad de zapatos como: botas, zapatillas, tenis, etc. El algoritmo se desarrolló en la plataforma de Matlab® con imágenes tomadas en un recorrido por las ciudades de Puebla y Cholula, los resultados obtenidos fueron que de un total de 46 imágenes procesadas, 41 fueron satisfactorias, 4 no satisfactorias y 1 falso positivo.

El algoritmo utiliza 5 zonas de riesgo para adaptarse a la velocidad del vehículo, utiliza el ventaneo para procesar una zona de riesgo de la imagen para buscar características de un zapato y determinar si hay una persona presente.

ABSTRACT

Today the number vehicles on a urban area increases rapidly, which the increased vehicles and people on the streets of the cities implies unfortunately increase traffic accidents, in México more than 17 thousand people die each year in this type of accidents and more than a third are pedestrians [1]. Driver distraction is the main cause of these accidents.

This thesis presents the development of an algorithm for detection of a person crossing transversely to the vehicle, specifically the shoe of the person who rest on the pavement, this provides a three level warning to an advanced driver assistance system in situations of risk that a pedestrian is present in the vehicle's way. Detection is performed by finding characteristic edges of a shoe with adaptive threshold to detect a wide variety of shoes like: boots, sneakers, tennis, etc. Algorithm was developed in Matlab platform with images taken on a tour of Puebla and Cholula cities, the results were that of total 46 images processed, 41 were successful, 4 unsatisfactory and 1 false positive.

The algorithm uses 5 risk zones to adapt to the vehicle speed, the algorithm uses windowing to process a zone of risk of the image to look for characteristics of a shoe and determine if there is a person present.

AGRADECIMIENTOS

Agradezco a todos mis amigos y amigas por su apoyo, que sin importar la distancia siempre me dieron ánimos de seguir adelante, a todos mis compañeros del instituto porque me apoyaron en todo lo que necesitaba.

Agradezco al Dr. Jorge Martínez Carballido por guiarme en el desarrollo de esta tesis resolviendo mis dudas en todo momento.

Agradezco al INAOE por las facilidades otorgadas durante mi formación académica y a todos los Profesores de Instrumentación por haberme aportado parte de su conocimiento.

Agradezco al CONACyT (Consejo Nacional de Ciencia y Tecnología) por el apoyo económico durante estos dos años de gran trabajo.

DEDICATORIA

Dedico esta tesis a todos mis familiares, a mis padres y mi hermano y hermana, que han sabido aconsejarme y guiarme durante todo este tiempo y siempre han sabido como animarme en aquellos días muy difíciles.

Gracias.

ÍNDICE GENERAL

Resumen	i
Abstract	iii
Agradecimientos	v
Dedicatoria	vii
Índice de Figuras	xiii
Índice de Tablas	xvii
Capítulo 1: Introducción	1
1.1 Motivación	2
1.2 Justificación	3
1.3 Problema	4
1.4 Objetivo	4
1.5 Hipótesis	4
1.6 Contribuciones	5
1.7 Estructura de la Tesis	5
Capítulo 2: Antecedentes	7
2.1 Puntos importantes en Sistemas PPS	8
2.2 Sistemas Comerciales	10
2.3 Sistemas Propuestos	14
2.3.1 Visión Monocular	14
2.3.2 Visión Estéreo	15

Capítulo 3: Detección del Objeto de Interés	17
3.1 Cámara fotográficas	18
3.2 Medición de distancias en una imagen	19
3.2.1 Calibración de la cámara	20
3.2.2 Relación distancia-pixeles de un zapato	22
3.3 Selección de la región de interés	23
3.4 Adaptabilidad de las zonas conforme a la velocidad	28
3.5 Selección del objeto de detección	30
3.5.1 Análisis del zapato	32
3.6 Formas de Detección	34
3.6.1 Binarización y detección de bordes	35
3.6.2 Umbral Adaptable	37
3.6.3 Binarización	38
3.6.4 Búsqueda de Bordes	39
3.6.5 Líneas vecinas	39
3.7 Binarización y búsqueda de bordes con umbral adaptable	40
3.7.1 Alineamiento de bordes	43
3.7.2 Límite de umbral	45
3.8 Aplicación en ventana	48
3.8.1 Desplazamiento de la ventana	48
3.8.2 Bordes Verticales	51
3.8.3 Bordes Horizontales	56
3.9 Decisión de Presencia de Zapato	61
3.10 Superposición de bordes en imagen original	61
3.11 Proceso alternativo subsecuente para presencia de un peatón	64
Capítulo 4: Resultados y Discusión	73
4.1 Resultados en imágenes con zapatos individuales	74
4.2 Resultados con el algoritmo completo	75
4.2.1 Positivos	77
4.2.2 Negativos	78

4.2.3 Falsos Positivos	78
4.2.4 Prueba en asfaltos	79
4.3 Discusión de Resultados	79
Capítulo 5: Conclusiones y Trabajo a Futuro	81
Conclusiones	81
Trabajo a Futuro	82
Bibliografía	83

ÍNDICE DE FIGURAS

Fig. 2.1: <i>Night Vision with Pedestrian Detection</i> de AUDI [5], [6], [7].	11
Fig. 2.2: <i>BMW Night Vision with pedestrian detection</i> [8], [9].	11
Fig. 2.3: <i>Night View Assist PLUS with Pedestrian Detection</i> de Mercedes-Benz [10], [11], [12].	12
Fig. 2.4: <i>Volvo Collision Warning with Full Auto Brake and Pedestrian Detection</i> [13], [14], [15], [16].	13
Fig. 3.1: Funcionamiento de una lente [24].	19
Fig. 3.2: Imágenes del primer objeto a distintas distancias.	20
Fig. 3.3: Imágenes del segundo objeto con 1 metro de ancho.	21
Fig. 3.4: Establecimiento de la región de interés para la detección de peatones. .	23
Fig. 3.5: Vista de la perspectiva de una carretera sobre una imagen.	24
Fig. 3.6: Vista de la perspectiva de la región de interés.	24
Fig. 3.7: Zonas de alarma de un sistema ADAS.	25
Fig. 3.8: Zonas de alarma adaptadas a la velocidad.	27
Fig. 3.9: Vista de perspectiva de las 5 zonas.	29
Fig. 3.10: Resultados de la detección de piel de la cara.	30
Fig. 3.11: Peatones que portan objetos diversos.	31
Fig. 3.12: Muestra de distintos tipos de zapatos.	32
Fig. 3.13: Muestra de dos diferentes calzados, a) pertenece a un hombre y b) a una mujer.	32
Fig. 3.14: Características similares en distinto tipos de zapatos.	33
Fig. 3.15: Bordos horizontales y verticales.	35

Fig. 3.16: Muestra numérica de la matriz numérica.....	36
Fig. 3.17: Diferencia de pixeles para los extremos del zapato.	36
Fig. 3.18: Diferencia de pixeles en un zapato blanco.	37
Fig. 3.19: a) Imagen Original, b) Binarización con las expresiones establecidas y c) aplicación de la función <i>sobel</i> de Matlab a la imagen.	38
Fig. 3.20: Borde de punta no lineal.....	39
Fig. 3.21: Binarización y búsqueda de bordes verticales.	41
Fig. 3.22: Binarización y búsqueda de bordes horizontales.	42
Fig. 3.23: diagrama del proceso completo para los cuatro bordes con umbral adaptable.....	46
Fig. 3.24: Ventana con zonas: crítica y no-crítica.....	49
Fig. 3.25: Desplazamiento de la ventana sobre la imagen.....	51
Fig. 3.26: Distintos casos probables de la posición del zapato en la ventana.	51
Fig. 3.27: Separación de 5 columnas entre el borde y la ventana.....	52
Fig. 3.28: Separación entre el borde de suela y la ventana.	54
Fig. 3.29: Diagrama de los procesos de bordes verticales con ventana.	55
Fig. 3.30: Ejemplo de bordes verticales de longitudes similares.	56
Fig. 3.31: Ejemplo de bodes disperejos conforme a la posición de la suela.	57
Fig. 3.32: Diagrama para los procesos de las dos horizontales en ventana.	60
Fig. 3.33: Condición para determinar la presencia de un Zapato.....	61
Fig. 3.34: Imagen final con los 4 bordes hallados superpuestos.....	64
Fig. 3.35: Imagen final mostrando los bordes hallados.	64
Fig. 3.36. Establecimiento de la separación para pierna.....	65
Fig. 3.37: Reposicionamiento de ventana para detección de pierna.	66
Fig. 3.38: Diagrama de los procesos para los bordes de pierna.	66
Fig. 3.39: Primera parte del Algoritmo Completo.....	70
Fig. 3.40: Segunda parte del Algoritmo Completo.....	71
Fig. 3.41: Tercera parte del Algoritmo Completo.....	72

Fig. 4.1: Comparación de dos imágenes para una misma zona.....	75
Fig. 4.2: Ejemplos de secciones de imágenes para distintas zonas.....	76
Fig. 4.3: Detección correcta con proceso de pierna en zona 4.	77
Fig. 4.4: Detección correcta en zona 5.....	77
Fig. 4.5: No detección en zona 5.....	78
Fig. 4.6: Detección no correspondiente a un zapato en zona 5.	78
Fig. 4.7: Asfaltos probados en el algoritmo.	79

ÍNDICE DE TABLAS

Tabla 3.1: Resultados del primer objeto con ancho de 24.2 cm.....	21
Tabla 3.2: Resultados del segundo objeto de la prueba.	21
Tabla 3.3: Relación de distancia-pixeles de dos zapatos.....	22
Tabla 3.4: Distancia de recorrido en metros a un segundo y dos segundos.	25
Tabla 3.5: Distancia de frenado de vehículos comerciales [28].....	26
Tabla 3.6: Distancia de detención conforme a la Ecuación 3.4.	27
Tabla 3.7: Establecimiento de zonas conforme a velocidad del vehículo.....	28
Tabla 3.8: Proceso de binarización y búsqueda para cada borde.....	43
Tabla 3.9: Aplicación del algoritmo para los cuatro bordes.	47
Tabla 3.10: Alto y ancho de la ventana para cada zona para una imagen con resolución de 6 Megapíxeles	48
Tabla 3.11: Separación entre verticales y filas para cada ventana para una imagen con resolución de 6 Megapíxeles	53
Tabla 4.1: Características del equipo utilizado para el desarrollo y prueba del algoritmo.....	73
Tabla 4.2: Resultados de imágenes de zapatos.....	74
Tabla 4.3: Resultados de las distintas imágenes aplicadas al algoritmo.....	76
Tabla 4.4: Tiempo relativo de ejecución para diversas imágenes procesadas	80

Capítulo 1

INTRODUCCIÓN

Desde el comienzo de la industria automotriz, a principios del siglo XX, los automóviles han pasado a formar parte de la vida cotidiana, usándolos para nuestro transporte personal y el de mercancías. Con el paso de los años han sido mejorados para ofrecer mayor rapidez, confort y seguridad.

Hoy en día, los automóviles cuentan con avanzados sistemas de seguridad tanto para proteger la vida de los tripulantes así como de la seguridad del exterior, tales sistemas como frenos antibloqueo (ABS), distribución electrónica de frenado (EBD), Control de Estabilidad (ESP), *Airbags* y chasis a prueba de deformaciones se encuentran cada vez en automóviles de media gama; pero hay sistemas más sofisticados que proporcionan asistencia al conductor para una mejor reacción ante casos de accidentes por circunstancias que el conductor no tiene control, por ejemplo, distracciones (llamada de celular, mirar a otro lado), cansancio, vista limitada, etc. En este tipo de asistencias conocidos como *ADAS (Advanced Driver Assistance Systems)* se encuentra establecido el sistema que trata esta tesis: la detección de peatones conocido como *PPS (Pedestrian Protection Systems)*.

El objetivo de un sistema *PPS* es detectar la presencia de personas en un área de interés específica alrededor del movimiento presente del vehículo con el fin de advertir al conductor.

Este tipo de sistema *ADAS* ya se encuentran en vehículos comerciales, pero denominados de gama alta y por tanto muy caros. Además, que en México desafortunadamente los autos que se venden la mayoría no cuenta con sistemas de seguridad “básicos” como ABS, Airbag para el conductor, etc. Y si disponen de estos sistemas por lo regular es en la versión más equipada y cara del modelo, además que de por si los autos vendidos en el país son más caros en comparación con otros mercados comparando el mismo modelo y versión de éste.

1.1 Motivación

Actualmente en nuestro país hay millones de automóviles circulando en las grandes ciudades y muchas veces los conductores no respetan las normas y señalamientos de tránsito, además que la mayoría de los vehículos no están tan equipados en cuestión de seguridad en comparación con los de países más desarrollados, y sobre todo, los sistemas más comunes están desarrollados para la protección de los tripulantes del vehículo excluyendo mayoritariamente a las personas que también usan la vía pública: los peatones.

Los sistemas *PPS* ayudan en este tipo de seguridad pero como se mencionó anteriormente, solo están disponibles en vehículos de gama alta y alto costo tales como de las marcas BMW, Audi, Volvo y Mercedes-Benz, y además que tienen sus limitaciones como el rango de detección y una velocidad limitada para que el sistema funcione.

Estos sistemas PPS montados en los vehículos comerciales usan más de un sensor para la detección, en mayoría, se recurren a sensores tipo, radar, de láser y cámara óptica, algunos emplean únicamente visión nocturna con infrarrojos.

Hay varios trabajos y artículos que proponen sistemas para detección de peatones y la mayoría se basa en imágenes, todos muestran buenos resultados pero lo que se detectó es que gran parte de estos trabajos proponen algoritmos complejos enfocándose a la detección del cuerpo entero o la mayoría de éste tanto que solo se tiene detección a personas cercanas además que las imágenes de sus pruebas son de dimensiones reducidas haciendo difícil determinar a qué velocidad y distancia son capaces de detectar. Otros emplean imágenes obtenidas por medio de cámaras tipo estéreo con lo cual se tiene un mejor rango de distancia para la detección.

1.2 Justificación

El automóvil ha venido a transformar la manera en que nos transportamos, nos facilita ir de un lugar a otro sin mucho esfuerzo, ayuda al transporte de mercancías a distancias largas.

De acuerdo a un estudio realizado por el INSP (Instituto Nacional de Salud Pública), a nivel mundial mueren 1.2 millones de personas en accidentes viales. En México más de 17 mil personas mueren al año por accidentes de tránsito y más de la tercera parte son peatones [1]. Estadísticamente, es la primera causa de muerte en jóvenes de 15 a 29 años y también de niños de 5 a 14 años de edad. Nueve de cada diez accidentes viales son producto de la conducta humana [2].

Un sistema que le brinde asistencia al conductor (*ADAS*), mejoraría considerablemente su reacción ante una eventualidad de riesgo, reduciendo el daño tanto a su persona como a terceros que estén involucrados en el riesgo, pero

estos sistemas son muy costosos de adquirir optando a que muchos usuarios de vehículos los consideren como un lujo innecesario e incluso no tengan conocimiento que existen dado que son sistemas no comunes.

1.3 Problema

Los sistemas PPS están disponibles para aquellas personas que tengan la capacidad monetaria de adquirir un vehículo de alta gama ya que éstos son los que disponen de este sistema. Además que tienen límites como la velocidad máxima del vehículo de por ejemplo 35 km/h (como el sistema de Volvo) y el número de sensores que utiliza, que elevan el costo del sistema.

1.4 Objetivo

Realizar un algoritmo orientado al hardware que sea capaz de realizar detección de peatones en una imagen donde el vehículo vaya a velocidades mayores de 35 Km/h, se tomó un límite de hasta 80 Km/h pero el algoritmo es capaz de tener mayor rango dependiendo de la resolución y calidad de la imagen.

1.5 Hipótesis

Para la detección se enfocó en la localización de los zapatos de la persona, dado que éstos tienen el menor grado de variación de forma, mediante la binarización y búsqueda de bordes con cierto umbral se determinan cuatro características principales:

- Verticales: talón y punta.
- Horizontales: suela y punta.

Si alguna de estas características no es hallada, el algoritmo buscará la pierna con dos verticales características correspondientes.

1.6 Contribuciones

Se propone un nuevo método para la detección de peatones en imágenes basado en un algoritmo orientado al hardware, es decir, se facilitará el traslado del algoritmo a un entorno de hardware con VHDL y así poder utilizar tarjetas FPGA para el desarrollo de un prototipo.

1.7 Estructura de la Tesis

Capítulo 2: Se explica los puntos principales para un sistema *ADAS* y las diferentes aplicaciones que existen en el mercado además de propuestas relacionadas con el tema.

Capítulo 3: Se detalla todo el proceso que se realizó para llegar a un sistema capaz de detectar el objeto de interés sobre una imagen.

Capítulo 4: Se muestran algunos ejemplos resultados y se da a conocer la eficacia del sistema conforme a las imágenes aplicadas.

Capítulo 5: Se concluye el presente trabajo y se enlistan futuras mejoras.

Capítulo 2

ANTECEDENTES

Dado al incremento de automóviles en el mundo y por ende el aumento de accidentes de tráfico, se ha motivado el desarrollo de vehículos inteligentes para salvaguardar la vida de sus usuarios desarrollando sistemas de ayuda al conductor (*ADAS*), pero enfocándose al conductor sin dar una importancia mayor al resto de usuarios de la carretera.

Para el desarrollo de sistemas de asistencia al conductor *ADAS* se deben cumplir algunos requisitos indispensables para que sea viable implementar un sistema confiable, entre los puntos fundamentales que se aplican para el desarrollo de un sistema de asistencia son:

- Sensores capaces de ofrecer una adecuada información en tiempo real del entorno donde se sitúa el vehículo.
- Una comprensión común acerca del comportamiento del vehículo en diferentes situaciones de riesgo y posibles actuaciones de minimización de riesgo.

- El sistema final que sea instalado en un vehículo comercial debe ser lo suficientemente robusto como para adaptarse a diferentes condiciones y cambios del entorno.
- Al tratarse de sistemas de seguridad críticos, deben ser altamente fiables. Por tanto, el proyecto debe cumplir todas sus fases de un modo exhaustivo y riguroso; desde los requisitos de la especificación del sistema, hasta el diseño, implementación y pruebas de funcionamiento.

Una vez puesto en claro qué requisitos deben cumplir este tipo de aplicaciones, cabe plantearse cómo llevar a cabo su desarrollo real. El alto nivel de automatización que se precisa en un vehículo autónomo, solo va a poder ser alcanzado una vez que el problema más desafiante de este ámbito sea resuelto: la percepción del entorno que rodea al vehículo.

2.1 Puntos importantes en Sistemas PPS

El desarrollo de sistemas de detección de peatones *PPS* es uno de los desafíos a cumplir dentro de los sistemas *ADAS*, esto es debido a la gran complejidad de factores que influyen para detectar la presencia de una persona, se mencionan los siguientes como los más significativos de acuerdo a [3] y [4].

- El escenario es complejo, los objetos aparecen en entornos muy variados y con gran cantidad de elementos, además que estos elementos aparecen en forma aleatoria y pueden estar en cualquier posición.
- Las condiciones de iluminación son cambiantes y desconocidas, y no se pueden controlar, ya que es propio del entorno exterior.

- Las personas pueden presentar una apariencia muy diversa, tanto al tamaño y postura del cuerpo además que visten ropa muy variada. Pueden darse situaciones en las que la ropa se confunda con el fondo de la imagen o bien la indumentaria de la persona puede salirse de lo considerado como normal (como llevar bolsos, gorras, sombreros, etc.).
- Los movimientos de peatones resultan impredecibles. En cualquier momento pueden variar de ritmo y/o trayectoria. Además, son objetos no-rígidos. Por otro lado, el peatón puede situarse con respecto a la cámara de múltiples maneras: de frente o con un ángulo, cambiando enormemente su apariencia.
- Además, las oclusiones y los grupos de personas añaden aún más complejidad a la tarea. También puede complicarse la tarea de detección si las personas llevan objetos como bolsas o carritos de la compra.
- La cámara está en movimiento, instalada en el vehículo. El propio movimiento de la cámara (ego-movimiento) dificulta sobremanera la tarea de la detección. Aunque cada vez hay equipos que compensan el movimiento propio de la cámara.
- Otro aspecto a considerar es la calidad de los sensores que directamente influye en las imágenes a procesar. Una baja calidad de los sensores, hace que se tengan imágenes malas y además se reduce el rango dinámico, lo que influye en su capacidad de adaptarse a cambios repentinos de iluminación y hace que aparezca una zona de la imagen más clara y otra muy oscura.

Todas estas peculiaridades del entorno de trabajo no se pueden controlar. Debido a la gran dificultad que supone desarrollar un sistema que funcione eficazmente en tales condiciones.

Actualmente ya se cuentan con sistemas comerciales, pero son pocos los vehículos que los equipan, también hay varios trabajos con sistemas propuestos para la detección de peatones, a continuación se detallan dichos sistemas tanto comerciales como propuestos.

2.2 Sistemas Comerciales

La introducción de Sistemas de Detección de Peatones en vehículos comerciales es muy reciente, de hecho apenas un poco más de un año, esto indica que realizar un sistema que detecte peatones en forma artificial es una tarea no fácil de realizar.

A continuación, se hace un análisis de los vehículos comerciales que ya cuentan con este sistema.

- **Audi:** Modelos A6 [5], A7 [6], A8 [7] año 2011, 2012: Estos vehículos tiene como opción de equipamiento un sistema de visión nocturna con detección de peatones (*Night Vision with Pedestrian Detection*) cuyas características son que utiliza imágenes infrarrojas y detecta la presencia de peatones y animales hasta 1000 pies (aproximadamente 300 metros) de distancia, tiene dos etapas, la primera es cuando detecta al peatón lo muestra en la pantalla de información al conductor resaltado de color amarillo. La otra etapa trata que si el objeto detectado se mueve dentro del camino del vehículo, el sistema muestra una alerta audible y sobresalta de color rojo la imagen en la pantalla. Tiene un precio entre 2,300 dólares (para modelos A8) y de 5,800 dólares (para modelos A6 y A7).



Fig. 2.1: *Night Vision with Pedestrian Detection* de AUDI [5], [6], [7].

- BMW:** Modelos de la Serie 5 [8] y Serie 7 [9] año 2010, 2011: dentro de su línea de equipamiento opcional, tienen un sistema de visión nocturna con detección de peatones y animales con una distancia hasta de 900 pies (aproximadamente 275 metros) usando tecnología infrarroja, donde se muestra un riesgo potencial en el camino en el monitor del sistema de navegación tal que el conductor pueda evitar el contacto antes de que sea visible a simple vista, en un rango muy cercano, se identifica la posición y la dirección a la que se está moviendo el objeto. Si un peatón se aproxima o cruza el camino, la tecnología de imagen termal alerta en la pantalla un símbolo en el panel de control y, si está equipado, con el sistema de visión sobre el parabrisas “*head-up display*”. Este equipamiento tiene un costo de 2,600 dólares para el sistema de visión nocturna con detección de peatones y 1,300 dólares para el “*head-up display*”.



Fig. 2.2: *BMW Night Vision with pedestrian detection* [8], [9].

- **Mercedes-Benz:** Clase S [10], CL [11], CLS [12] año 2010, 2011: Estos modelos cuentan con un sistema de asistencia nocturna de serie (salvo la clase S que es opcional) denominado *Night View Assist PLUS with Pedestrian Detection* que hace el uso de tecnología infrarroja, puede detectar objetos hasta 500 pies de distancia (aproximadamente 150 metros) en total oscuridad. El sistema no solo detecta objetos calientes como animales, sino también escombros fríos y autos estacionados, éstos son mostrados en una imagen blanco y negro sobre la pantalla de alta resolución situada en el panel de instrumentos. Un software avanzado ayuda a detectar formas humanas hasta 300 metros de distancia para luego resaltarlas sobre la pantalla. Como equipamiento opcional tiene un costo de 1,780 dólares.



Fig. 2.3: *Night View Assist PLUS with Pedestrian Detection* de Mercedes-Benz [10], [11], [12].

- **VOLVO:** Modelos XC60 [13], XC70 [14], S60 [15], S80 [16] año 2010, 2011: Cuenta con un paquete de seguridad denominado *City Safety* de segunda generación, el cual contiene un sistema para prevención de colisión contra otro vehículo y un sistema de detección de peatones, con ambos sistemas tienen la particular de realizar frenado automático cuando no se detecta reacción por parte del conductor. El sistema de detección de peatones automáticamente evita la colisión a una velocidad de hasta 22 mph

(aproximadamente 35 km/h). El conductor primeramente es alertado mediante una alarma sonora y una alerta visual que se muestra en el parabrisas, si el conductor no responde ante la alarma y el sistema evalúa que la colisión está muy cercana, los frenos del vehículo son aplicados con fuerza total. El auto solo frena si es demasiado tarde para aplicar una maniobra evasiva y aplica los frenos en menos de un segundo antes del tiempo calculado de impacto. El sistema usa una combinación de sensores tipo radar y una cámara para identificar peatones parados o en movimiento dentro de un campo de visión de 60° a luz de día. Tiene la limitante de que la cámara solo observa en el espectro visible (visión humana). Tanto como el radar y la cámara tienen un campo de visión limitado y pueden ser afectados por lluvia, nieve, brisa o luz baja de visibilidad (sol bajo). A pesar de ser un sistema novedoso, la compañía tiene la siguiente advertencia: “Si la distancia inicial del objeto fue muy cercana, la advertencia puede llegar demasiado tarde mientras que la función del sistema de soporte puede ser no garantizada. El conductor siempre es encargado de la responsabilidad”. Este sistema tiene un costo de 2,700 dólares.

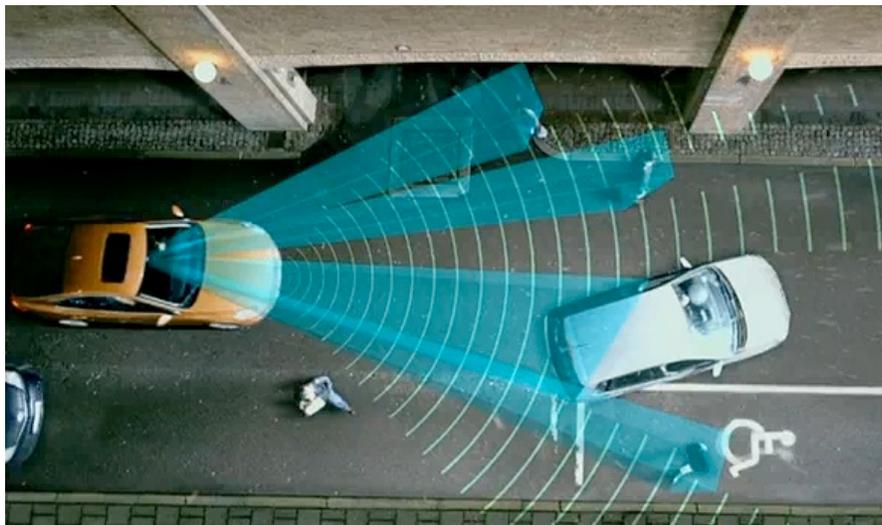


Fig. 2.4: Volvo Collision Warning with Full Auto Brake and Pedestrian Detection [13], [14], [15], [16].

Como se observa, estos sistemas no están disponibles en todos los modelos de una misma marca, incluso modelos más costosos que los descritos no tiene como equipamiento opcional de seguridad este tipo de sistemas.

2.3 Sistemas Propuestos

Dado que en los últimos años se aumentó el interés sobre los sistemas de detección de peatones, el número de propuestas para este tipo de sistemas también ha aumentado, en forma general hay dos clasificaciones en general de acuerdo al sistema de visión de la cámara que se utiliza:

- **Visión Monocular:** usan solo un tipo de cámara para la captura de imágenes.
- **Visión Estéreo:** usan dos tipos de cámara para imitar en todo lo posible la visión de los ojos humanos, con este tipo de visión es posible la percepción de profundidad, se deben usar cámaras de las mismas características.

De acuerdo al tipo de visión que utiliza cada propuesta, se presentan dos secciones explicando en cada una las propuestas que utilizan ese tipo de visión.

2.3.1 Visión Monocular

Una de las primeras propuestas de detección de peatones fue la de Papageorgiou y Poggio [17], que hace el uso de Haar wavelets y un clasificador SVM (*Support Vector Machine*) donde se calcula la diferencia de pixeles entre dos áreas rectangulares en diferentes configuraciones, pero solo lo hace para detección en imágenes de una cámara estática. Surgió en el año 2000.

Las propuestas para un sistema orientado a un vehículo se dieron en el mismo año 2000, una es dada por D.M. Gavril [18], que denomina a su detector como “*the Chamfer System*” basado en la detección de objetos basados en la forma. Esto lo hace usando una variedad de la transformada de distancia denominada “transformada chamfer” para obtener el contorno del peatón y mediante plantillas previamente clasificados hace la comparación para determinar la presencia del peatón en una imagen.

Shashua et al [19], este trabajo trata de la clasificación de partes del cuerpo basadas en características basadas en gradiente de la imagen. Usando un clasificador basado en la integración de la representación de imágenes locales, se reduce la variabilidad de cada parte local que tomando una forma completa donde se encuentren todas las partes y reduce las regiones de interés mediante clasificadores *SVM*, cada región se divide en trece partes superpuestas. Las salidas de los clasificadores se alimentan como las normas de menor peso a una máquina *AdaBoost* que establece la regla de clasificación final.

2.3.2 Visión Estéreo

Otra propuesta es de Liang Zhao y Charles E. Thorpe [20]. Realizaron un detector basado en redes neuronales en imágenes estereo, usando un mapa de disparidad y el Laplaciano de la Transformada Gaussiana separa los objetos tanto del primer plano (*foreground*) como los del fondo (*background*) y mediante el gradiente de magnitud de la imagen realizan una red neuronal *feed forward* para descartar los objetos que no estén en la dimensión normal de una persona que por lo regular son los objetos del fondo.

Broggi et al [21], usan la comparación de siluetas, donde un modelo binario de la forma de la parte superior del cuerpo es comparado con un módulo de imagen de bordes por una simple correlación después de una segmentación basada en simetría.

Dariu M. Gavrilă [22]. En su trabajo mediante una secuencia de imágenes se obtiene un “árbol” de plantillas a nivel de contornos de formas humanas y se hace una comparación con otras imágenes mediante una jerárquica de formas de menor a mayor nivel de detalle.

C. Hilario et al [23], en su propuesta usan primeramente un mapa de disparidad proveniente de una imagen estéreo y en las partes filtradas donde hay una simetría vertical alta aplican un modelo de contornos activos conocidos como “*snakes*”.

De las tres últimas propuestas, la similitud con el algoritmo desarrollado es la detección de bordes, donde se difiere, es el método para la detección de dichos bordes y el tipo de visión, dado que se usará una sola cámara (monocular).

Para tener un mayor conocimiento de los últimos sistemas propuestos, se recomienda revisar el estudio realizado por David Gerónimo et al [4], donde se hace una revisión de casi todos los trabajos de los últimos 10 años, con los dos distintos tipos de visión y diferentes tipos de sensor (como infrarrojos) y describe la forma en que se hace la clasificación del objeto de interés.

Capítulo 3

Detección del Objeto de Interés

En los puntos que se mencionan en el capítulo 2 como los factores a considerar en un sistema PPS, primeramente se planteó qué características debe considerarse en una imagen, tras el análisis de varias imágenes tomadas en una situación real se llegaron a las siguientes conclusiones:

- Resolución: a mayor resolución de la imagen hay mayor detalle, es posible tener un mayor rango de distancia de detección.
- Profundidad: dado a que entre más lejos esté el objeto de interés, éste será de menor dimensión en la imagen, es decir, será más pequeño en la imagen.
- Iluminación: dado a que un entorno urbano tiene tanta variedad, el contraste entre sombras, zonas oscuras y partes claras es un factor que puede afectar seriamente el desempeño del sistema.

3.1 Cámaras fotográficas

Se dispusieron de dos cámaras para la adquisición de las imágenes, una es de la marca SONY modelo DSLR-A290Y, la otra cámara es de la marca CANON modelo PowerShot SD1200, a detalle, se especifican a continuación las características de cada cámara.

Tabla 1: Características de las cámaras fotográficas utilizadas.

	Sony DSLR-A290Y	CANON
Resolución	14.2 MPx	10 MPx
Estabilización de imagen	Si: SteadyShot INSIDE®	Si: Estabilizador Óptico
Sensibilidad	Hasta ISO 3200	Hasta ISO 1600
No. De imágenes por segundo	Máximo 3 por segundo	Hasta 3 por cada 2 segundos
Distancia focal.	18 – 55 mm	6.2 – 18.6 mm

La Cámara Sony Pertenece al INAOE, por tanto solo en contadas ocasiones se tuvo disposición de dicha cámara, ésta principalmente se usó para pruebas de control con distancias conocidas en el estacionamiento perteneciente al INAOE; con la que mayormente se trabajó fue con la cámara CANON a una resolución de 6 megapíxeles que se usó para la toma de imágenes en un recorrido real sobre un automóvil en la ciudad de Puebla. La resolución se eligió para tener mayor detalle aunque no es un factor crucial para el funcionamiento del sistema pudiéndose elegir otra resolución.

3.2 Medición de distancias en una imagen

En [24], usan una técnica para medir la distancia a la que está localizado un objeto en una imagen, para ello se basan mediante la formación de la imagen proyectada (conocida como imagen virtual) del objeto de interés en el plano de la imagen.

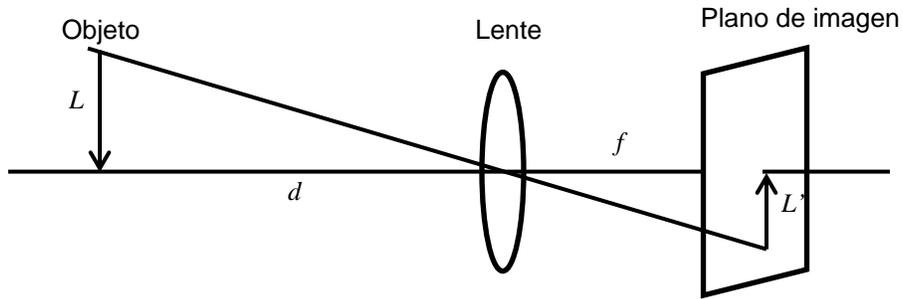


Fig. 3.1: Funcionamiento de una lente [24].

En Fig. 3.1, f es la distancia de la lente al plano de la imagen, conocida también como distancia focal, d es la distancia de la lente al objeto, L' es el tamaño proyectado en el plano de la imagen y L es el tamaño real del objeto. De acuerdo a la fórmula óptica en [24], la distancia d es inversamente proporcional al tamaño de la imagen virtual L' .

$$d = \frac{L}{L'} f \dots\dots\dots \text{Ecuación 3.1}$$

Dado que L' es la medida virtual del objeto dada por un producto entre el número de píxeles n por un factor de conversión α , se expresa como sigue:

$$L' = n\alpha \dots\dots\dots \text{Ecuación 3.2}$$

3.2.1 Calibración de la cámara

Para poder obtener los parámetros f y α es necesario calibrar la cámara con el procedimiento que se explica en el capítulo 3 de [24], mediante el uso del *Camera Calibration Toolbox for Matlab*, que consta de tomar una serie de fotografías a un tablero de ajedrez determinando la longitud de f . De la Ecuación 3.1 y Ecuación 3.2 se sigue el procedimiento detallado en el capítulo 4 del mismo autor.

Los resultados para los parámetros requeridos son:

$$f = 37.3914 \text{ mm}$$

$$\alpha = 0.1385$$

Para prueba de control, se realizaron tomas de imágenes de dos objetos con ancho conocido a intervalos de un metro, el primero tiene un ancho de 24.2 cm y el otro tiene 1 m de ancho. Para conocer los pixeles que ocupa el objeto, las imágenes se tomaron a una distancia conocida, entonces los parámetros L y d son conocidos dejando como incógnita la variable n , de la Ecuación 3.1 se despeja n para obtener el número de pixeles:

$$n = \frac{L}{d\alpha} f \dots\dots\dots \text{Ecuación 3.3}$$

Para el primer objeto se hicieron tres tomas, además que está resaltado de un color para que se distinga fácilmente:



Fig. 3.2: Imágenes del primer objeto a distintas distancias.

Aplicando la Ecuación 3.3 se obtuvieron los siguientes resultados:

Tabla 3.1: Resultados del primer objeto con ancho de 24.2 cm.

Imágenes	Píxeles				
	1 metro	2 metros	3 metros	4 metros	5 metros
1ra. Toma	643	335	226	172	132
2da. Toma	641	337	230	174	140
3ra. toma	640	333	226	172	138

Para el caso del segundo objeto, primeramente se hicieron cálculos para determinar el número de píxeles, después se realizó la prueba con la Ecuación 3.3 y se compararon resultados.



Fig. 3.3: Imágenes del segundo objeto con 1 metro de ancho.

Los resultados son los siguientes:

Tabla 3.2: Resultados del segundo objeto de la prueba.

Distancia en metros	Calculado	Medido en Matlab
1	2700 píxeles	2744 píxeles
2	1350 píxeles	1419 píxeles
3	900 píxeles	948 píxeles
4	675 píxeles	715 píxeles
5	540 píxeles	571 píxeles
6	450 píxeles	477 píxeles
7	386 píxeles	401 píxeles
8	337 píxeles	359 píxeles
9	300 píxeles	318 píxeles
10	270 píxeles	286 píxeles

3.2.2 Relación distancia-píxeles de un zapato

Teniendo la base de cómo medir la distancia y los píxeles de un objeto sobre una imagen, se realizaron pruebas en el estacionamiento del instituto con la cámara Sony acondicionada a 6 MPx a distancias controladas que van desde los 5 hasta los 50 metros con intervalos de 5 metros con dos zapatos distintos y de diferentes medidas, uno es de 29 cm y el otro de 24 cm. Los resultados tras verificarlos son:

Tabla 3.3: Relación de distancia-píxeles de dos zapatos.

Distancia en metros	Píxeles	
	<i>29 cm</i>	<i>24 cm</i>
5	245	200
10	125	100
15	80	65
20	60	50
25	50	40
30	40	35
35	35	28
40	30	24
45	27	22
50	25	20

Con ello ya se tiene una base para establecer la distancia máxima de detección en la región de interés, se nota que el zapato disminuye su tamaño conforme la distancia es mayor, esto se debe tener en cuenta para una correcta aplicación del algoritmo sobre la imagen.

3.3 Selección de la región de interés

Previamente para la búsqueda de un objeto de interés, se debe definir la zona donde se realizará dicha búsqueda conocida como *Región de interés*, dado que la percepción de un impacto de alto riesgo es al frente de la trayectoria del vehículo, es posible descartar partes de la imagen que no son importantes para la búsqueda del objeto.

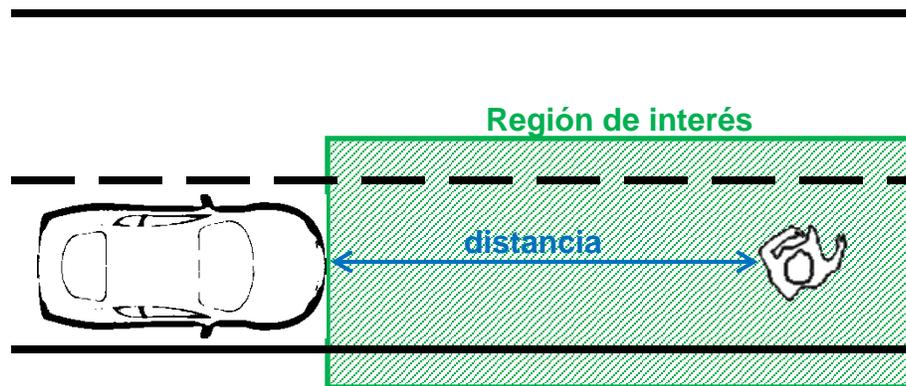


Fig. 3.4: Establecimiento de la región de interés para la detección de peatones.

Como se muestra en la Fig. 3.4, La región de interés solo consta del abarcamiento del carril donde transita el automóvil dado que es la zona donde cruzan transversalmente los peatones con respecto al auto y por ende, mayor riesgo de accidente, pudiendo excluir así zonas como el carril subyacente para la detección.

De acuerdo al Manual de Proyecto Geométrico de Carreteras en México [25], el ancho aceptado por carril para una carretera es de 3.65 metros, tomando este dato como referencia, el ancho de la región de interés será un poco mayor a la medida del carril ya que, pensando en un entorno urbano, las calles son más anchas que un carril de carretera y por ello se cubre un área mayor, otro dato importante son las dimensiones de vehículos comerciales, los auto y camionetas de pasajeros no exceden los 2.2 metros de ancho.

En una imagen la perspectiva hace que las laterales paralelas de la carretera conforme se avanza en profundidad, se acerquen más una a la otra, teniendo una visión de la carretera en forma trapezoidal o triangular.

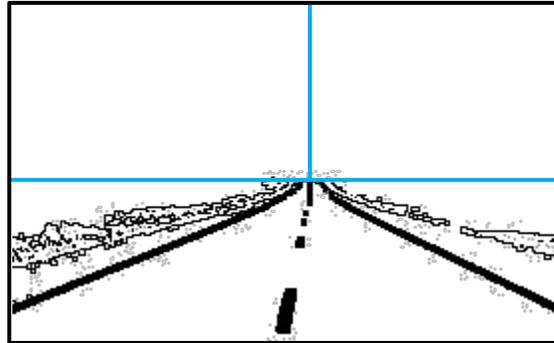


Fig. 3.5: Vista de la perspectiva de una carretera sobre una imagen.

Hay que tomar en cuenta que influye la posición de la cámara, su ángulo de visión y la altura a la que está dispuesta ya que esto puede afectar el punto de intersección entre la posición de la perspectiva (línea azul vertical) y la profundidad (línea azul horizontal), por lo regular se desea que este punto de intersección esté lo más centrado posible en la imagen.

Pero como se mencionó anteriormente, para el propósito de este proyecto solo se tomará en cuenta el carril donde circula el automóvil para efectos de detección, de manera que se mostraría de la siguiente forma:

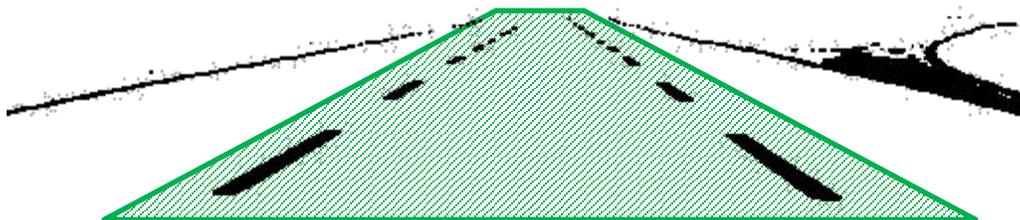


Fig. 3.6: Vista de la perspectiva de la región de interés.

Los sistemas *ADAS* por lo regular presentan tres zonas de riesgo para tres niveles de alarma clasificadas como Alto, Medio y Bajo y se caracterizan por ser zonas constantes y fijas.



Fig. 3.7: Zonas de alarma de un sistema *ADAS*.

Pero para este proyecto se toma en cuenta la velocidad del vehículo, ya que es factor de acuerdo a la cantidad de distancia que recorre antes que el conductor realice una acción evasiva, conforme a [25], [26], y [27] hay una regla para estar a una distancia segura llamada “Regla de los dos segundos” la cual establece como dos segundos de separación entre dos vehículos para evitar una colisión sin importar a la velocidad que se circule.

Tabla 3.4: Distancia de recorrido en metros a un segundo y dos segundos.

Km/h	m/s	m. 2 seg.
20	5.56	11.11
25	6.94	13.89
30	8.33	16.67
35	9.72	19.44
40	11.11	22.22
45	12.50	25.00
50	13.89	27.78
55	15.28	30.56
60	16.67	33.33
65	18.06	36.11
70	19.44	38.89
75	20.83	41.67
80	22.22	44.44

La *Distancia de Frenado* de un vehículo, es la distancia que tarda desde que se acciona el freno hasta que se detiene completamente el vehículo, esta distancia varía conforme a cada vehículo, el tipo, el tamaño, su peso, etc., en [28] se muestran los resultados de pruebas a varios vehículos comerciales.

Tabla 3.5: Distancia de frenado de vehículos comerciales [28].

Vehículo	Velocidad	Distancia
Tsuru	80 km/h	35 m
Sentra	80 km/h	38 m
Ren. Sandero	80 km/h	29 m
Fiesta	80 km/h	28 m
Jetta	80 km/h	27 m
Corolla	80 km/h	31 m
Peg 206	80 km/h	28 m
Aveo LT	80 km/h	25 m
CrossFox	80 km/h	30 m
Cruze	80 km/h	28 m
Fiat Punto	80 km/h	24 m
Honda Fit	80 km/h	28 m

Además hay otro factor que influye mucho para la detención del vehículo, es el tiempo que le tarda al conductor en asimilar el riesgo y tomar una acción al respecto llamado *Tiempo de Reacción*, este tiempo varía de acuerdo a la persona, su edad, su condición física, su estado (drogado, alcoholizado), etc. Varios estudios como en [25] indican que va desde 0.5 segundos hasta 2.5 segundos en el peor de los casos, y en [26] indican que una persona en estado normal va desde 0.5 hasta 1 segundo. De acuerdo al Código de Circulación de España, se aplica una fórmula para determinar la distancia máxima a la que se deberá detener el vehículo tomando en cuenta los dos factores anteriores.

$$d = \frac{v^2}{100} \dots\dots\dots \text{Ecuación 3.4}$$

Donde d es la *Distancia de Detención* y v es la velocidad del vehículo. Aplicando la fórmula para distintas velocidades se obtiene la siguiente tabla.

Tabla 3.6: Distancia de detención conforme a la Ecuación 3.4.

Km/h	Distancia en metros
20	4
30	9
40	16
50	25
60	36
70	49
80	64
90	81
100	100

Teniendo en cuenta los estudios anteriores, las zonas de alarma se adaptarán de acuerdo a la velocidad del vehículo, de forma que se desplazarán a mayor distancia conforme la velocidad sea más alta.



Fig. 3.8: Zonas de alarma adaptadas a la velocidad.

Esta adaptabilidad de las zonas se debe a que a mayor velocidad del vehículo, mayor distancia por cada segundo recorre, por tanto no es de gran ayuda que el sistema detecte a un peatón cerca del vehículo cuando se circula a altas

velocidades, por ejemplo, si se circula a 80 km/h y se detecta un peatón a 15 metros de distancia, sólo se tendrá 0.68 segundos para reaccionar y tomar una acción evasiva, conforme al tiempo de reacción quizás incluso no se reaccione siquiera cuando se aviste al peatón. Claro que es difícil ver que una persona se cruce muy cerca cuando hay vehículos a altas velocidades, esto es un factor que no se controla y es muy difícil dar una solución ante una situación como esta.

3.4 Adaptabilidad de las zonas de riesgo conforme a la velocidad

De acuerdo a lo estipulado sobre la medición de distancia y la relación de distancia-objeto en la imagen, es necesario adaptar las zonas de alarma de acuerdo a la velocidad que va el vehículo, tomando como referencia los datos de Tabla 3.3 y Tabla 3.6 se estipularon 5 corrimientos de zonas donde cada una cubre una determinada velocidad y distancia.

Tabla 3.7: Establecimiento de zonas conforme a velocidad del vehículo.

Velocidad en KPH	Zona	Distancia en metros
10 – 20	1	5 – 9 m
21 – 40	2	10 – 19 m
41 – 50	3	20 – 29 m
51 – 60	4	30 – 39 m
61 – 80	5	40 – 50 m

La zona que corresponde conforme a la velocidad es la zona más cercana de riesgo la cual es la **Zona Roja** dado a que se desea una pronta detección del riesgo de colisión con el peatón, de forma general, las zonas estarían estipuladas en la región de interés como se observa en la Fig. 3.9.

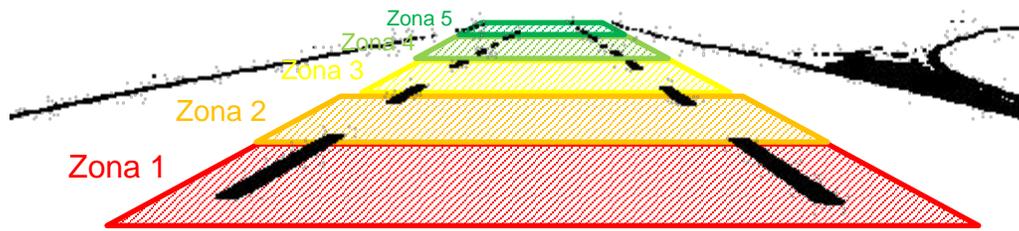


Fig. 3.9: Vista de perspectiva de las 5 zonas.

Las 5 zonas están estipuladas para una reacción óptima del conductor al momento de la alarma, además con el supuesto de que el peatón sigue su curso; varios estudios muestran que la velocidad de un peatón promedio va de 4 a 6 Km/h [29], [30], [31], es decir, de 1.12 a 1.67 m/s, claro que es un factor que depende de la edad, la condición física, discapacidad, etc. del peatón, pero tomando esos datos, al peatón le tomaría de 2 a 3 segundos en cruzar el carril donde circula el vehículo.

De lo anterior se desprende que siguiendo el supuesto que el peatón sigue su curso, el vehículo no necesariamente necesita detenerse totalmente, basta con que reduzca buena parte de su velocidad para permitir al peatón el cruce completamente el carril.

A partir de lo anterior, se tiene las bases para determinar en qué parte de la imagen se realizará la detección de la presencia del peatón.

3.5 Selección de objeto de detección

Debido a la gran variedad de las personas en las imágenes, tanto en su posición, vestimenta, tamaño, etc. Se analizaron varias imágenes para seleccionar un objeto en común en los peatones para su detección, ante una primera revisión se optó por la detección del color de piel suponiendo que la mayor parte visible está en la cara, En [32] se detalla un algoritmo para la detección de piel para *RGB*:

```
R > 95 AND G > 40 AND B > 20 AND max{R, G, B} - min{R, G, B} > 15 AND
    % RGB components must not be close together greyness elimination
|R - G| > 15 AND
    % also R and G components must not be close together -
    % otherwise we are not dealing with the fair complexion
R > G AND R > B
    % R component must be the greatest component
OR
    % The skin colour under flashlight or (light) daylight
    % lateral illumination⇒
R > 220 AND G > 210 AND B > 170 AND |R - G| ≤ 15 AND R > B AND G > B
    % R and G components must be close together
    % B component must be the smallest component
```

Aplicando este algoritmo a imágenes seccionadas de la cabeza se obtuvieron los siguientes resultados:



Fig. 3.10: Resultados de la detección de piel de la cara.

Se observa que en imágenes claras el algoritmo tiene un correcto funcionamiento, pero en imágenes en sombra casi no hay detección, con el inconveniente de que aquellos colores que estén en el rango del algoritmo serán detectados como piel, como en este caso el color naranja de la gorra y el color café de la chamarra, esto se debe a que el algoritmo toma en cuenta todas las tonalidades de piel desde la más clara hasta la más oscura, además de la suposición que el peatón está de perfil dado a que su cruce es transversal por tanto, el área de piel es reducido en la cara influyendo también factores como la vestimenta como sombreros, bufandas, pasamontañas, etc., las sombras o la posición del cabello, por tanto para un entorno urbano común esta técnica quedó descartada.

Analizando otra vez las imágenes y seccionando solamente donde hay presencia de peatones, se pensaba aplicar mediante plantillas la comparación de formas preestablecidas de peatones, pero se observó que hay una gran variedad de las posturas y vestimenta de los peatones como faldas, abrigos, sombreros, etc., y en varios casos hay presencia de objetos que discrepan de la forma habitual de la pose de un peatón que camina transversalmente, tales objetos son mochilas, bolsas, cajas, etc. Por tanto se necesitaría una base de datos extensa para cubrir todas las posibles situaciones y sobretodo se **complicaría el algoritmo** teniendo la necesidad de una gran capacidad de hardware, ante esta conclusión, se descartó la técnica dado que uno de los objetivos principales es la minimización en hardware.

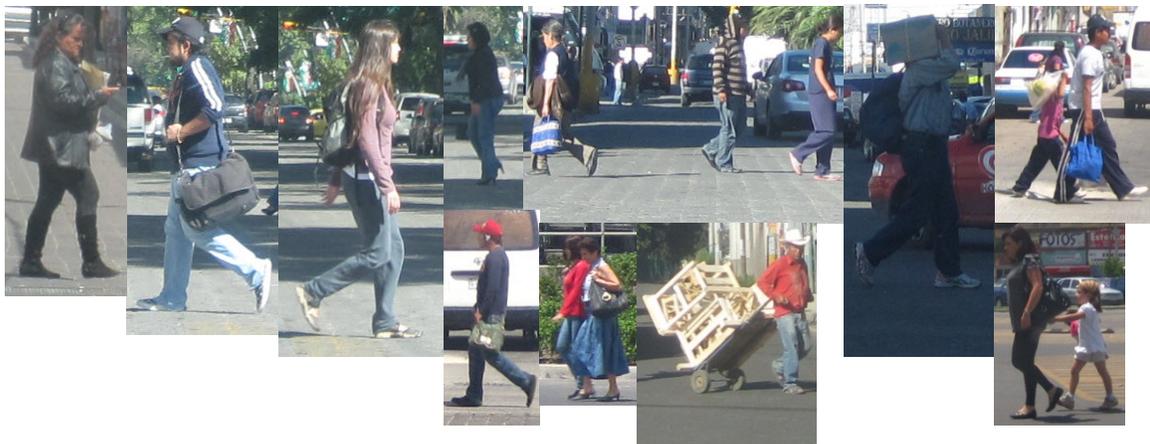


Fig. 3.11: Peatones que portan objetos diversos.

Al ver todas las imágenes, de todas las características comunes que definen a una persona como lo son: cabeza, cuerpo, brazo y piernas, y analizando cuál de todas éstas tiene menos variación y menos obstaculización, se encontró que la zona de los pies es la que menor variación tiene. Por tanto se optó por **el zapato como el objeto de detección** en la imagen.



Fig. 3.12: Muestra de distintos tipos de zapatos.

3.5.1 Análisis del zapato

Seleccionando el zapato como objeto de interés, revisando distintas imágenes de personas y observando el mercado de zapatos, se halla que hay una extensa variedad de zapatos, tanto en diseños, tamaños, colores, etc. además si están orientados a una actividad, (trabajo, deporte, gala, etc.), además considerar que un zapato de hombre difiere de un zapato de mujer por las características físicas propias de cada uno, tanto en tamaño como en diseño.



a)



b)

Fig. 3.13: Muestra de dos diferentes calzados, a) pertenece a un hombre y b) a una mujer.

Como diferencias principales en cada zapato, se pueden enlistar las siguientes:

- **Tacones:** tal como se muestra en la Fig. 3.13, en zapato femenino es posible encontrar tacones altos lo cual cambia el aspecto normal del zapato.

- **Punta:** es donde se halló gran diversidad, como punta alta, punta baja, punta de pico, punta chata, en a) puede clasificarse como punta alta y chata y b) como punta baja y de pico, por tanto pueden darse varias combinaciones en la punta del zapato, esto es dado por el diseño propio del zapato.
- **Altura del zapato:** hay zapatos con suela baja y suela alta, comúnmente llamados “de piso” y “plataforma”, donde se encuentra mayor variedad es en el zapato femenino, dado que los tacones aumentan la altura trasera del zapato, pero la punta difiere de esa altura siendo mucho más baja tal como se muestra en b) de Fig. 3.13, en zapato masculino es más uniforme la altura sin importar si es alta o baja como se muestra en a) de Fig. 3.13.

A pesar de todas esas diferencias tras el análisis de los distintos tipos de zapatos tanto para hombre como mujer, hay 3 características principales que están presentes en cualquier zapato: **Talón, Punta y Suela.**



Fig. 3.14: Características similares en distinto tipos de zapatos.

Como se aprecia en la Fig. 3.14, se ven remarcado en tres colores las características de los zapatos, de color amarillo indica la punta, de color verde claro el talón y de color rojo la suela. Notándose que para efecto de detección esta serán las características que se deben hallar indistintamente del tipo de zapato.

3.6 Forma de detección

Teniendo definido el objeto de interés a detectar, el siguiente paso es definir el método para su detección. Hay muchos métodos para la detección de objetos sobre una imagen, en la revisión de propuestas en el Capítulo 2, cada trabajo presenta un método distinto con sus distintas ventajas y desventajas [4], si se desea hacer con plantillas para comparación, la gran variedad de estilos de zapatos llevará a tener una extensa base de datos. De usarse una técnica que requiera aprendizaje mediante técnicas existentes: *wavelets* o redes neuronales, necesitará gran capacidad de cómputo para que el sistema “aprenda” una situación nueva cada vez que se presente, recordando que se estipula aplicar en tiempo real. Pero uno de los objetivos es que el sistema sea económico, por tanto aplicar alguno de los métodos anteriores sería contraproducente.

3.6.1 Binarización y detección de bordes

Tomando muestras de las imágenes de zapatos, y al estar en color *RGB*, es decir, la imagen está compuesta por 3 matrices distintas una por cada color, se eligió solo una matriz que en este caso fue la matriz *R*, esto es dado que para la detección de bordes no es necesario que la imagen esté a color eligiendo solo una matriz la imagen se comporta como si fuera en escala de grises y en forma general la matriz es un arreglo numérico que indica la intensidad del color correspondiente a cada pixel.

Para detectar los bordes de las características principales de los zapatos, hay que estructurar qué tipo de bordes son los que se adecúan a los zapatos. Tras observar todas las muestras de zapatos, se concluyó que hay 4 bordes comunes, 2 bordes verticales que corresponden a los extremos de punta y el talón, y 2 bordes horizontales que corresponden a la suela y la parte superior de la punta.



Fig. 3.15: Bordos horizontales y verticales.

De la Fig. 3.15 se observa que las líneas naranjas corresponden a la punta y el talón que son consideradas como verticales, y las líneas rojas corresponden a la suela y la parte superior de la punta consideradas como horizontales.

Teniendo definido los tipos de bordes a detectar, se prosigue a desarrollar la forma para detectarlos, Matlab ya dispone de varios filtros para detección de bordes ya sea por métodos como el laplaciano, gradiente, canny, Roberts, etc. Además que éstas se aplican usando máscaras sobre la imagen, las cuales ya está definidas por el propio método utilizado.

La diferencia es que se aplicó una expresión de comparación para poder binarizar la imagen y al mismo tiempo discriminar datos que no son de interés, dado que es muy posible que se presente ruido en una imagen que pueda afectar para la decisión de la presencia de borde, analizando la matriz numérica de la imagen, se encontró que mediante una simple diferencia entre dos pixeles es posible obtener el borde del objeto de interés, y la discriminante es la comparación con un umbral que solo tomará como positivos aquellas diferencias que cumplan la condición.

$$b(x, y) = [|a(dif(x))| > um] + [|a(dif(y))| > um] \dots\dots\dots \text{Ecuación 3.5}$$

El umbral um está definido por el segundo pixel de la diferencia multiplicado por un valor numérico, esto es dado que la diferencia de pixeles siempre es variable y por tanto es necesario adaptar el umbral a cada diferencia entre pixeles.

16	118	115
17	161	113
18	139	97

Fig. 3.16: Muestra numérica de la matriz numérica.

Pero hay una consideración a tomar en cuenta, observando dónde estarán los posibles bordes en las imágenes se halló que la diferencia de los pixeles de la punta es inversa a la diferencia de pixeles del talón, lo mismo sucede para el caso de las horizontales.



Fig. 3.17: Diferencia de pixeles para los extremos del zapato.

Tal como se observa en la Fig. 3.17, la diferencia de pixeles es inversa de entre ambos extremos de verticales, lo mismo se aplica para la diferencia de pixeles para las horizontales, ante tal situación se establecieron dos expresiones para cada tipo de borde, en total son 4 expresiones una correspondiente para cada uno de los cuatro bordes estipulados.

$$b_1(r, c) = |[a(r, c) - a(r, c + 1)]| > [a(r, c + 1) * u] \dots\dots\dots \text{Ecuación 3.6}$$

$$b_2(r, c) = |[a(r, c + 1) - a(r, c)]| > [a(r, c) * u] \dots\dots\dots \text{Ecuación 3.7}$$

$$b_3(r, c) = |[a(r, c) - a(r + 1, c)]| > [a(r + 1, c) * u] \dots\dots\dots \text{Ecuación 3.8}$$

$$b_4(r, c) = |[a(r + 1, c) - a(r, c)]| > [a(r, c) * u] \dots\dots\dots \text{Ecuación 3.9}$$

Donde la Ecuación 3.6 y Ecuación 3.7 corresponden a las verticales de punta y de talón respectivamente y la Ecuación 3.8 y Ecuación 3.9 a las horizontales punta

superior y suela respectivamente. En cada una de las expresiones r corresponde a la fila de la matriz y c a la columna de la misma.

La diferencia se tiene como valor absoluto dado a que puede ser negativa, este caso se presenta cuando el zapato es de color más claro que el pavimento, por ejemplo, un tenis de color blanco.

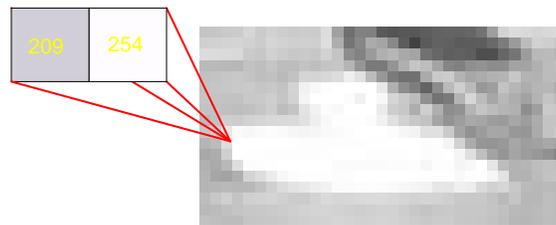


Fig. 3.18: Diferencia de pixeles en un zapato blanco.

3.6.2 Umbral adaptable

De la Ecuación 3.6 a la Ecuación 3.9, se tiene que el umbral está determinado por el segundo término de la diferencia multiplicado por un factor numérico u , este factor numérico no es constante.

Para determinar el rango del umbral se analizaron varias matrices R de distintas imágenes de zapatos desde oscuras a claras, y conforme a la diferencia de los dos pixeles para la binarización se estableció que la diferencia debe ser al menos entre 5% y un 20% mayor al segundo término, por tanto, u estará en el rango de 0.05 a 0.20. Pero este valor se aplica de mayor a menor, es decir, la diferencia primeramente debe ser mayor al 20% dado a que mayoritariamente los casos el zapato contrasta más que el pavimento, si no hay un buen resultado, se reduce paulatinamente hasta el mínimo valor del 5%.

$$um = a \left(\begin{matrix} r, m \\ m=c, c+1 \end{matrix} \right), a \left(\begin{matrix} n, c \\ n=r, r+1 \end{matrix} \right) * u \Big|_{0.05}^{0.2} \dots\dots\dots \text{Ecuación 3.10}$$

El rango del factor está estipulado para disminuir en lo posible el ruido en la binarización pero que no se descarten casos donde el zapato tiene un color cercano al pavimento, en zonas oscuras como en sombra o incluso en imágenes no tan nítidas.

3.6.3 Binarización

Puesto que son 4 bordes los de interés, primeramente se binarizará para las dos verticales, sabiendo que en los extremos de la imagen es donde se encuentran los bordes, entonces para cada expresión no es necesario aplicarla sobre toda la imagen.

Para efectos de prueba, las expresiones para las verticales solo se aplicarán de cada extremo hasta la mitad de la imagen. Lo mismo se aplica para las horizontales, esto es simplemente para verificar el funcionamiento de las expresiones.

Mediante Matlab® se realizó un programa para los 4 bordes, además que la misma imagen se le aplicó la función *sobel* propia del software, en la siguiente figura se muestran las comparativas.

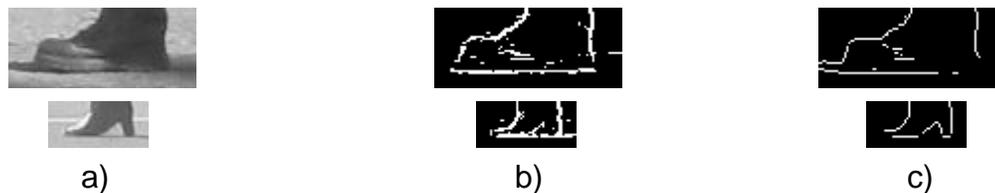


Fig. 3.19: a) Imagen Original, b) Binarización con las expresiones establecidas y c) aplicación de la función *sobel* de Matlab a la imagen.

Tal como se observa en la figura anterior, la binarización muestra que es suficiente para la detección, dado a que la aplicación de la función *sobel* es incierto en el resultado final y puede que contenga información que no sea de interés como en el caso de la imagen c) superior que contiene el borde de la sombra, o se elimine información que es fundamental como en la imagen c) inferior donde no se muestra la vertical que corresponde a la punta.

3.6.4 Búsqueda de Bordes

Una vez que se tiene binarizada la imagen, se necesita determinar la presencia de los bordes característicos de un zapato, pero antes es necesario establecer la longitud mínima de cada borde que además depende de la zona en donde se localice el zapato.

Otra circunstancia es la posición del zapato, es decir, que lo primero que se presente es la punta o el talón del zapato, por tanto para discriminar que borde se halla primero (en una búsqueda de izquierda a derecha) y tras analizar el caso más remoto: zapato en la zona más lejano con la punta más delgada, la longitud mínima de borde vertical es de 2 píxeles.

3.6.5 Líneas vecinas

Un último factor es el hecho de que los bordes en la binarización no son completamente rectos tal como se muestra en la siguiente figura.



Fig. 3.20: Borde de punta no lineal.

Para saber la longitud total se hace una búsqueda de líneas vecinas, es decir, no solo se busca la primera línea con longitud mínima de 2 píxeles sino además se buscan en la segunda y tercera columna donde se halló la primera línea, la longitud total es la posición del inicio de la primera línea y la posición del final de la tercera línea. A continuación se muestra el pseudocódigo para búsqueda de bordes, aplica para todos los bordes:

```
Para inicio to fin filas
  Para inicio to fin columnas
    Si hay dos píxeles = 1
      Guarda inicio
      Cuenta píxeles seguidos = 1 hasta hallar un 0
      Guarda fin 1
      Buscar en columna(ver) o fila(hor) siguiente
      Si se halla pix = 1
        Cuenta pix seg = 1 hasta hallar un 0
        Guarda fin 2
        Buscar en columna(ver) o fila(hor) siguiente
        Si se halla pix = 1
          Cuenta pix seg = 1 hasta hallar un 0
          Guarda fin 3
```

3.7 Binarización y búsqueda de bordes con umbral adaptable

Es posible asociar los procesos de binarización y búsqueda para cada borde y así no es necesario que se realice el proceso completo para poder ejecutar el siguiente, además que el umbral para un borde no sea el mismo para otro dado por la variedad propia de la imagen como se había explicado antes (sombras, posición, formas, etc.), por tanto al poder binarizar y buscar a la vez se evita el procesamiento de toda la imagen además que es posible activar banderas para la binarización y búsqueda de los bordes conforme se halle el anterior.

De esta manera es posible cambiar el umbral si no se haya el borde deseado pudiendo ser diferente para cada borde, esto puede pasar por ejemplo cuando el zapato esté a mitad de sombra o que tenga varios colores (como los tenis por ejemplo) y además que si no se halla el borde no se procede al proceso del segundo borde. Para tener indicios de la presencia de un zapato, los dos bordes verticales deben estar presentes, se estableció así dado a que la separación de ambos bordes es lo que determina si corresponde a las medidas de un zapato, si no se hallan estos dos bordes no se procede para cualquiera de los dos bordes horizontales. De manera gráfica, el proceso para los bordes verticales es:

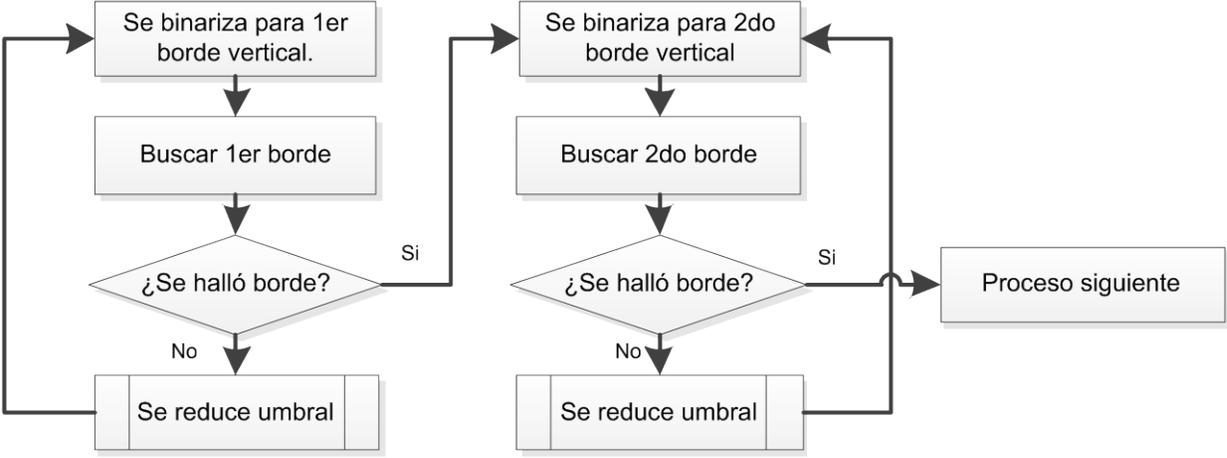


Fig. 3.21: Binarización y búsqueda de bordes verticales.

Tal como se muestra en la Fig. 3.21, es necesario encontrar el primer borde para poder encontrar el segundo borde y además el umbral disminuye si no hay un indicio claro del borde buscado, por lo tanto hace que se tenga un umbral diferente para cada borde teniendo 2 umbrales en total.

Para la búsqueda de los bordes horizontales, al igual que los bordes verticales, es necesario hallar uno para continuar con el siguiente, primeramente se empieza con la horizontal de la punta, si se halla, se procede para la horizontal de suela. Al

definir la longitud mínima para cada horizontal, similarmente obtenidas del mismo caso que de los bordes horizontales, se tiene que para la horizontal punta, al menos debe haber 2 pixeles y para la horizontal suela 5 pixeles, ésta última es dado a que es el borde que más consistencia tiene, es decir es el más lineal de todos, pero también se aplica la búsqueda de líneas para ambos bordes horizontales para obtener un mejor resultado y cada borde tiene su propio umbral. De manera gráfica, el proceso para los bordes horizontales es:

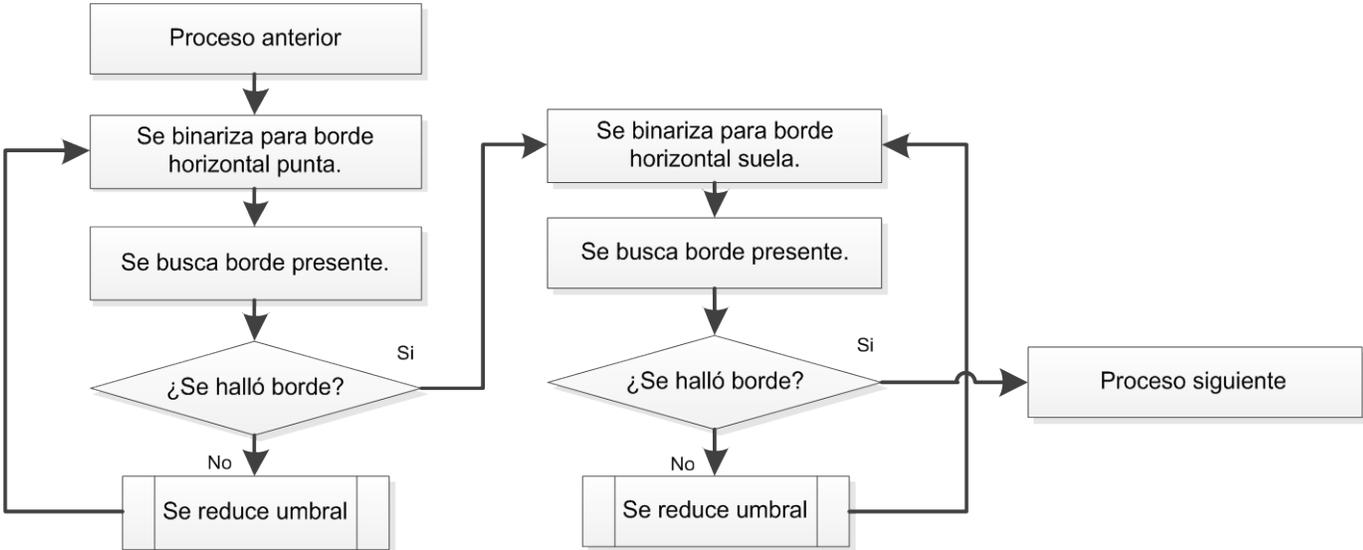


Fig. 3.22: Binarización y búsqueda de bordes horizontales.

Los bordes horizontales se buscan en las filas y columnas guardadas donde se hallaron los bordes verticales dado a que no es necesario buscar en toda la imagen puesto que en donde se hallaron los bordes es donde está la mayor probabilidad de hallar los bordes horizontales, el borde horizontal de punta está determinado por el inicio del borde vertical de menor longitud el cual debe corresponder a la vertical de punta, el borde de suela es determinado por el fin de ambos bordes verticales dado a que no hay mucha diferencia en dónde terminan los bordes verticales.

Al aplicar los procesos de binarización y búsqueda para cada borde, se obtienen los siguientes resultados:

Tabla 3.8: Proceso de binarización y búsqueda para cada borde.

Primera vertical	Segunda vertical	Primera horizontal	Segunda horizontal
			
			

3.7.1 Alineamiento de bordes

Para propósitos de presentación, se pueden alinear los bordes encontrados y mostrarlos sobre una imagen, asegurando así el correcto funcionamiento. Dado a que ya se tienen almacenados los índices de principio y fin de los bordes encontrados, simplemente comparando cuál es el índice menor de inicio y el índice mayor de fin, recordando que se buscan dos líneas vecinas más, se traza el borde mostrando así la longitud real encontrada para cada uno de los bordes. Se muestra a continuación el pseudocódigo de comparación:

Parte inicial

Si 1er borde, 2do borde y 3er borde hallado

 Si inicio_b1 < inicio_b2 & inicio_b3

 Ind_inicio = inicio_b1

 Si no

 Si inicio_b1 < inicio_b2 & > inicio_b3

 Ind_inicio = inicio_b3

 Si no

 Si inicio_b1 > inicio_b2 & < inicio_b3

 Ind_inicio = inicio_b2

```

        Si no
            Si inicio_b2 < inicio_b3
                Ind_inicio = inicio_b2
            Si no
                Ind_inicio = inicio_b3
Si no
    Si 1er borde & 2do borde hallado
        Si inicio_b1 < inicio_b2
            Ind_inicio = inicio_b1
        Si no
            Ind_inicio = inicio_b2
    Si no
        Si 1er borde & 3er borde hallado
            Si inicio_b1 < inicio_b3
                Ind_inicio = inicio_b1
            Si no
                Ind_inicio = inicio_b3
Parte final
Si 1er borde, 2do borde y 3er borde hallado
    Si fin_b1 > fin_b2 & fin_b3
        Ind_fin = fin_b1
    Si no
        Si fin_b1 > fin_b2 & < fin_b3
            Ind_fin = fin_b3
        Si no
            Si fin_b1 < fin_b2 & > fin_b3
                Ind_fin = fin_b2
            Si no
                Si fin_b2 > fin_b3
                    Ind_fin = fin_b2
                Si no
                    Ind_fin = fin_b3
Si no
    Si 1er borde & 2do borde hallado
        Si fin_b1 > fin_b2
            Ind_fin = fin_b1
        Si no

```

```
        Ind_fin = fin_b2
Si no
    Si 1er borde & 3er borde hallado
        Si fin_b1 > fin_b3
            Ind_fin = fin_b1
        Si no
            Ind_fin = fin_b3
```

Una vez que se obtiene el índice de principio y fin para cada borde, simplemente se traza una línea sobre la imagen con dichos índices, verificando así el funcionamiento de los procesos en la imagen.

3.7.2 Límite de umbral

En los proceso de binarización y búsqueda el umbral se reduce cada vez que no hay un resultado de búsqueda satisfactorio, pero si se hace esta operación recursivamente habrá un momento en que el resultado sea negativo afectando así todo el proceso dando resultados no deseados, para evitar esta contrariedad, recordando que los límites del umbral está entre el 5% y 20% del segundo término de la diferencia, se introducen estos límites en el proceso de reducción de umbral comparando cada vez si se ha llegado al límite, al unir los procesos para los cuatro bordes con los límites de umbral, se obtiene lo siguiente:

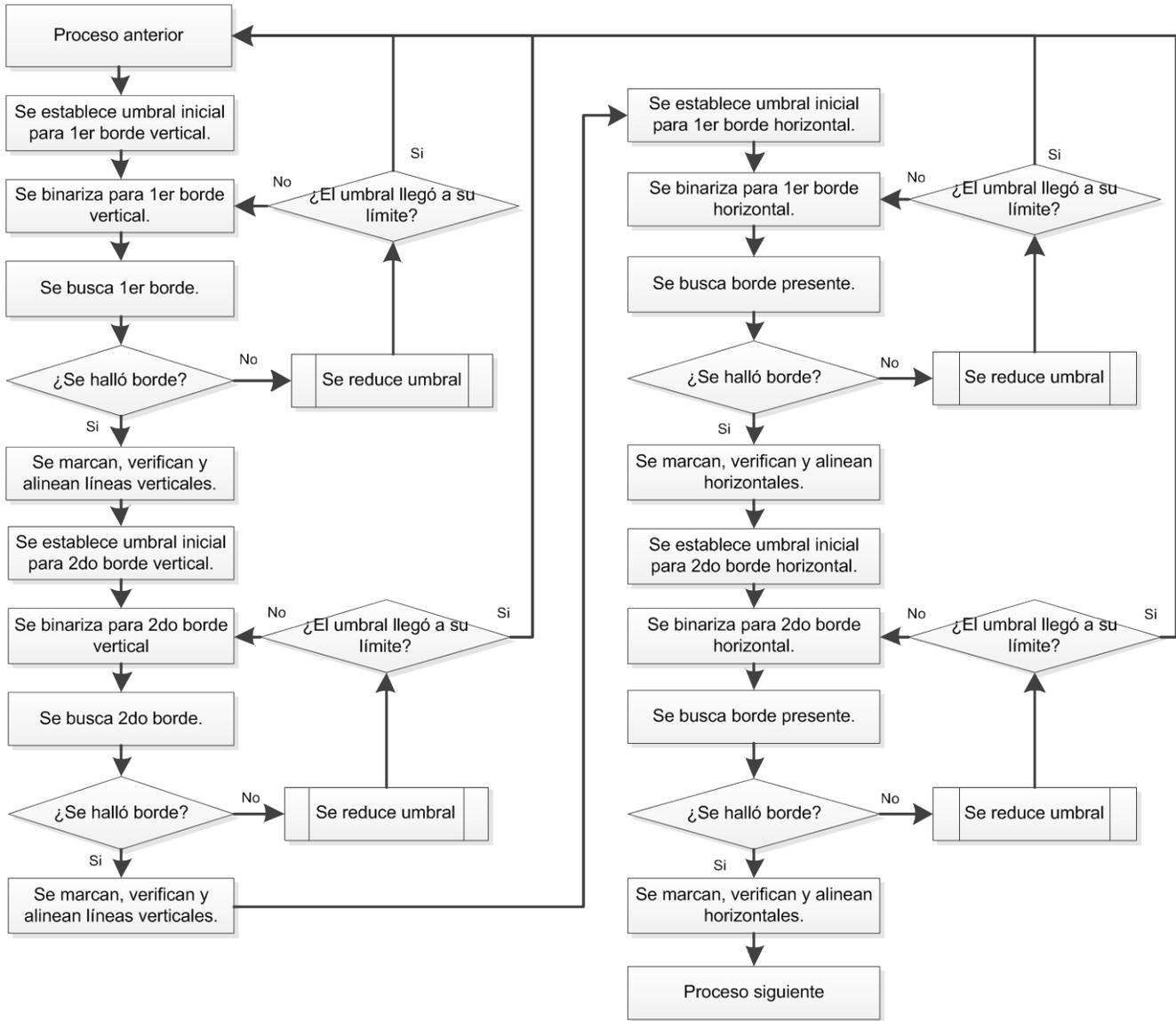


Fig. 3.23: diagrama del proceso completo para los cuatro bordes con umbral adaptable.

En la figura anterior se muestra lo que es el “corazón” del algoritmo, dado que es aquí donde se hace el procesamiento fundamental para la detección de las características fundamentales de un zapato, más adelante se detallarán los procesos anteriores y posteriores al algoritmo para poder aplicarlo en imágenes de un entorno real.

Tabla 3.9: Aplicación del algoritmo para los cuatro bordes.

Primera vertical	Segunda vertical	Primera horizontal	Segunda horizontal
			
			
			
			

De la tabla anterior se observa el alineamiento de los cuatro bordes, dejando claro el funcionamiento de los procesos de binarización, búsqueda y alineamiento, aunque solo se muestran resultados de dos zapatos, se hicieron pruebas con más tipos de zapatos, esto se mostrará en el capítulo 5 de Resultados. Lo importante es que el procedimiento funciona correctamente y el siguiente paso es la aplicación de todo el algoritmo en imágenes conforme a las distintas zonas ya establecidas, donde hay que adecuar todo el algoritmo para que tenga un correcto funcionamiento.

3.8 Aplicación en ventana

Tal como se había analizado antes, en una imagen de interés hay 5 zonas establecidas de acuerdo a la velocidad del vehículo, por tanto, es necesario definir las medidas para cada ventana, cómo se desplazará la ventana y además las adecuaciones para el algoritmo en cuanto a cómo se aplicará en dicha ventana teniendo en cuenta la distancia entre bordes verticales y las limitaciones necesarias para un correcto funcionamiento.

Puesto que ya se tienen las medidas de ciertos zapatos (Tabla 3.3) y sabiendo la distancia que cubre cada zona (Tabla 3.7), las medidas de cada ventana para que cubra ambos aspectos fueron establecidas para que cubra las medidas de los zapatos que cubre cada zona, quedando de la siguiente manera:

Tabla 3.10: Alto y ancho de la ventana para cada zona para una imagen con resolución de 6 Megapíxeles.

Zona	Alto	Ancho
1	140	280
2	70	140
3	35	70
4	25	50
5	20	40

3.8.1 Desplazamiento de la ventana

Puesto que la ventana tiene un alto y ancho predefinido y la imagen igual tiene un número de filas y columnas totales mucho mayor que la ventana. La forma en que se desplazará la ventana sobre la imagen está conformada por un número de iteraciones tanto para filas como para columnas, puesto que se puede saber la

dimensión de la imagen de la zona correspondiente y las dimensiones de la ventana mediante una operación simple se obtiene el número de iteraciones para cada caso.

```
Iter_fil = alto_imagen/alto_ventana;  
Iter_col = ancho_imagen/ancho_ventana;
```

Dado a que los números de las iteraciones pueden ser decimados, puede redondearse al número entero más cercano o simplemente usando datos del tipo entero que solo contienen número sin decimales.

Otro inconveniente es que la última iteración, que accede a las últimas filas y columnas, éstas no sean suficientes para que la dimensión de la ventana pueda abarcar esa sección de la imagen, por tanto los lazos de las iteraciones tendrán una iteración menos para evitar ese inconveniente.

```
Para iter1 de 1 to Iter_fil - 1  
  Para iter2 de 1 to Iter_col - 1  
    ...  
    ...
```

Las dimensiones de la zona de la imagen están calculadas para que incluyan una zona no-crítica donde la cual la ventana al no pasar en dicha zona por las iteraciones descartadas, no se influya en el funcionamiento para la detección del zapato en la región de interés, en la Fig. 3.24 se muestra sombreado en amarillo la zona no-crítica.



Fig. 3.24: Ventana con zonas: crítica y no-crítica.

El inicio de la ventana será desde las primeras posiciones de las columnas y las últimas posiciones de las filas, es decir, de abajo hacia arriba puesto que se necesitar detectar al peatón frente al peligro más cercano y la parte inferior de la imagen es la zona más cercana de impacto.

```
%% Filas
inicio_fil = alto_imagen - alto_ventana + 1
fin_fil =alto_imagen
```

```
%% Columnas
inicio_col = 1
fin_col = ancho_ventana
```

La ventana abarcará parte de la imagen conforme a sus dimensiones, para la siguiente iteración abarcará ahora la suma de sus dimensiones tanto en inicio más fin de la posición de las columnas de la ventana anterior, pero en filas será un retroceso ya que la primera posición fue en las últimas filas de la imagen.

```
Para iter1 de 1 to Iter_fil - 1
    Para iter2 de 1 to Iter_col - 1
        ...
        ...

        col_ini_ventana1 = col_ini_ventana1 + ancho_ventana
        col_fin_ventana1 = col_fin_ventana1 + ancho_ventana

    Fin iter1

    fil_ini_ventana1 = fil_ini_ventana1 - alto_ventana
    fil_fin_ventana1 = fil_fin_ventana1 - alto_ventana
    col_ini_ventana1 = 1
    col_fin_ventana1 = ancho_ventana

Fin iter2
```

Si se observa el pseudocódigo anterior, primeramente se tiene una posición de filas y el recorrimiento se hace sobre todas las columnas, al terminar la última iteración de columnas se hace el siguiente recorrimiento de filas y se reinician las posiciones de las columnas dado que se necesita recorrer otra vez la ventana sobre toda la “franja” en esa posición de filas, tal como se muestra en Fig. 3.25:



Fig. 3.25: Desplazamiento de la ventana sobre la imagen.

3.8.2 Bordes Verticales

La diferencia que habrá entre la aplicación del algoritmo por medio de una ventana radica en que dado el recorrimiento que hará la ventana por la imagen, para la búsqueda de la primera vertical no se limitará por una porción de la ventana que contendrá una sección de la imagen, sino que se hará la búsqueda en toda la ventana dado que no hay ninguna certeza de que el zapato esté establecido enteramente en la ventana, es más probable que solo se tenga una porción de dicho zapato y además esté en una posición en donde sea difícil la detección de los demás bordes como por ejemplo estar seccionado el zapato.



Fig. 3.26: Distintos casos probables de la posición del zapato en la ventana.

Se muestra en la Fig. 3.26 algunos casos de los más probables que sucedan a la hora de aplicar la ventana, para afrontar estas dificultades, se propuso que al momento que se halle la primera vertical, se reacomode la ventana de tal forma que abarque el resto del zapato y así poder buscar los bordes restantes.

Para cuando los casos de que el hallazgo del primer borde vertical esté muy al extremo derecho de la imagen y no sea posible cubrir el segundo borde vertical, para evitar esta dificultad, dado que se conoce la posición de la columna del borde hallado, es posible desplazarlo hacia el extremo izquierdo con una separación de al menos 5 columnas más ante la primera columna, es decir, la posición final del primer borde vertical será en la columna 5 de la ventana, siempre y cuando cumpla con una condición dada ya que no es necesario desplazarla cuando el borde está sobre el primer tercio del ancho de la ventana.

```
Si col_borde_1 > [1/3(ancho_ventana) - 5]
    col_ini_ventana2 = col_borde_1 + col_ini_ventana1 - 5;
    col_fin_ventana2 = col_borde_1 + col_fin_ventana1 - 5;
Si no
    col_ini_ventana2 = col_ini_ventana1;
    col_fin_ventana2 = col_fin_ventana1;
```

Con lo anterior se asegura que el segundo borde se incluya para la siguiente búsqueda sin importar en dónde esté colocado el zapato en la zona, dado que la venta es suficientemente grande para cubrir el zapato más cercano de la zona. Además la separación es la misma para todas las ventanas.



Fig. 3.27: Separación de 5 columnas entre el borde y la ventana.

Para los casos donde el primer borde esté muy arriba o muy abajo sobre la ventana y no se contemplen los bordes horizontales, se estableció que deben ser desplazadas las filas obteniendo que el zapato esté más “centrado” sobre la ventana, para ello se fijaron dos variables más para las ventanas, una variable indica la separación que debe haber al menos entre las dos verticales para que el menor de los casos (zapato más lejano de la zona) sea cubierto, y la otra variable indica una separación mínima que debe haber entre la última posición de la fila en donde se halló el primer borde vertical (de arriba hacia abajo) y la última fila correspondiente de la ventana (también de arriba hacia abajo), éste ajuste debe ser suficiente para que el zapato esté lo más acomodado posible en la ventana para posteriores búsquedas de bordes.

Tabla 3.11: Separación entre verticales y filas para cada ventana para una imagen con resolución de 6 Megapíxeles.

Ventana de zona	Separación entre verticales en columnas	Separación de filas
1	90	20
2	50	15
3	30	10
4	20	5
5	15	5

Al igual que el acomodo en columnas para incluir el segundo borde vertical, se realiza un acomodo en filas para asegurar la inclusión de los dos bordes horizontales.

```
fil_fin_ventana2 = fil_ini_ventana1 + fil_fin_borde_1 + sep_filas
fil_ini_ventana2 = fil_fin_ventana2 - alto_ventana
```

Los valores de inicio y fin de las filas difieren de la forma en que se realiza para las verticales, dado a que la separación más importante es entre el borde de suela y la última fila de la ventana por ello primero se hace la operación para adecuar el fin de filas de la ventana, y simplemente se realiza una operación simple para adecuar el inicio de las filas puesto que el alto de la ventana siempre es el mismo, no hay necesidad de hacer una comparación ya que las verticales establecieron las posiciones necesarias para incluir el zapato total, solo es necesario mover las filas como se muestra en Fig. 3.28.



Fig. 3.28: Separación entre el borde de suela y la ventana.

Con los procedimientos anteriores se elimina cualquier caso que dificulte la detección de los bordes sobre la ventana, primeramente para la búsqueda de los bordes verticales, como mencionó anteriormente, el primer borde vertical se busca sobre toda la ventana por los casos mencionados, al reacomodar la ventana la búsqueda del segundo borde ya no se realiza enteramente en la ventana, sino se hace sobre una porción de ésta ya que se tiene conocido las direcciones de filas y columnas del 1er borde y la separación mínima que debe haber sobre el 2do borde, por tanto al reacomodar toda la ventana solo es necesario que el proceso del segundo borde se realice en una porción de la ventana proporcionada por la separación de verticales.

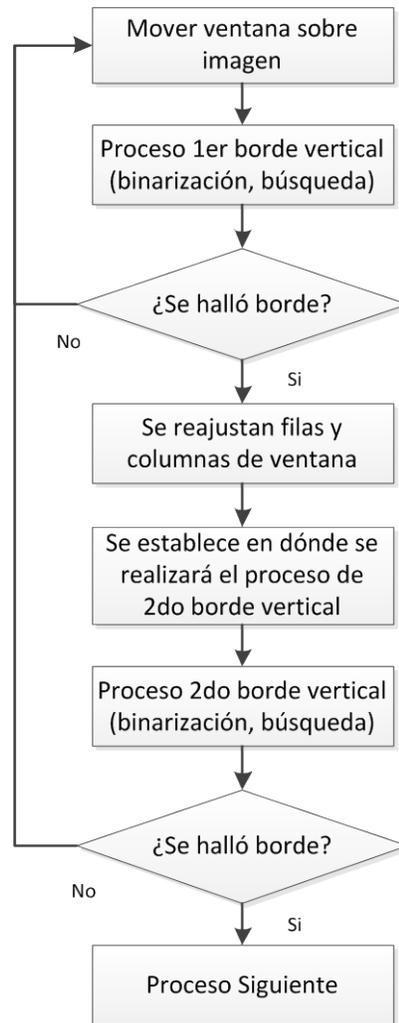


Fig. 3.29: Diagrama de los procesos de bordes verticales con ventana.

Como se aprecia en la Fig. 3.29, solo el proceso de la primera vertical debe realizarse en toda la ventana, para cuando se halle se reajusta la posición de la ventana y sabiendo la separación que hay entre verticales, se hace el proceso en esa porción de la ventana además que si no es hallado alguno de los dos bordes la ventana se desplaza a la siguiente posición de la imagen para comenzar una nueva búsqueda de bordes.

3.8.3 Bordos Horizontales

Cuando son hallados los dos bordes verticales y se reacomoda la ventana se hace el procedimiento para la detección de los bordes horizontales, primeramente se verifica que la separación entre los bordes verticales sea lo suficiente para considerarse que pertenecen a un zapato y ello activa una bandera para que los procesos siguientes se ejecuten.

```
%% Cuando el 1er borde es hallado en el extremo derecho de la ventana
Si col_borde_1 > [1/3(ancho_ventana) - 5]
    Si (col_borde_2 - 5) > sep_columnas
        proc_sig = 1;
    Si no
        proc_sig = 0;
%% Cuando el 1er borde está sobre el límite permitido de la ventana
Si no
    Si (col_borde_2 - col_borde_1) > sep_columnas
        proc_sig = 1;
    Si no
        proc_sig = 0;
```

Cuando se hallan los bordes verticales se conocen y guardan sus ubicaciones, con ello es posible establecer en dónde se realizarán los procesos para los bordes horizontales, tal como se mencionó anteriormente, la longitud de cada borde vertical es distinta habiendo un borde más largo que otro, donde el más largo corresponde al talón y el otro a la punta, pero hay casos donde la detección de los dos bordes no sea correspondiente conforme a un talón y a una punta, es decir, que se hallen los dos bordes pero con una longitud similar que no sea distinguible cuál pertenece a talón y cuál a punta como se aprecia en la Fig. 3.30.

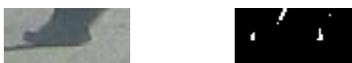


Fig. 3.30: Ejemplo de bordes verticales de longitudes similares.

Además, puede que la posición del fin de los dos bordes no sea similar, es decir, dado a que la suela es el borde más lineal, se espera que ambos bordes verticales terminen muy cercano a las filas donde se encuentre el borde de suela pero puede haber casos en que algún borde vertical no termine cerca de estas filas puede que esté desplazado hacia arriba como se muestra en la Fig. 3.31.



Fig. 3.31: Ejemplo de bodes disperejos conforme a la posición de la suela.

Por tanto, para evitar tales dificultades, se realiza un proceso de decisión con los datos guardados de las posiciones de los bordes para indicar en qué posiciones se realizarán los procesos de los bordes horizontales.

Si los bordes son similares en longitud y están situados en las mismas filas, el proceso de horizontal punta se realizará en todas las columnas que hay entre los dos bordes. Para que se considere que un borde es mayor que el otro debe tener al menos 4 pixeles más que el otro borde para que se considere perteneciente a talón.

Si los bordes están desplazados, es decir, unos está más arriba que el otro, los procesos de las dos horizontales se realizará sobre la posición de filas del borde que esté más abajo, el borde de horizontal punta será en el inicio de filas y el borde de horizontal de suela en el fin de filas de dicho borde vertical. La propia condición de los 4 pixeles mayor que el otro borde lo realiza dado que siempre se compara el inicio de los dos bordes (de arriba hacia abajo) y por ende el que sea de menor fila de inicio indica que su posición es más arriba que el otro borde.

```
Si fil_ini_borde1 < fil_ini_borde2
    fil_hor_punta = fil_ini_borde2
    fil_hor_suela = fil_fin_borde2
```

```

Si |fil_ini_borde2 - fil_ini_borde1| > 3
    check_pant1 = 1
    Si col_borde1 > [1/3(ancho_ventana) - 5]
        col_hor1_ini = [1/2(5 + col_borde2)]
        col_hor1_fin = col_borde2
        col_hor2_ini = 5
        col_hor2_fin = col_borde2
    Si no
        col_hor1_ini = [1/2(col_borde1 + col_borde2)]
        col_hor1_fin = col_borde2
        col_hor2_ini = col_borde1
        col_hor2_fin = col_borde2
Si no
    check_pant2 = 1
    Si col_borde1 > [1/3(ancho_ventana) - 5]
        col_hor1_ini = 5
        col_hor1_fin = col_borde2
        col_hor2_ini = 5
        col_hor2_fin = col_borde2
    Si no
        col_hor1_ini = col_borde1
        col_hor1_fin = col_borde2
        col_hor2_ini = col_borde1
        col_hor2_fin = col_borde2
Si no
    fil_hor_punta = fil_ini_borde1
    fil_hor_suela = fil_fin_borde1
    Si |fil_ini_borde1 - fil_ini_borde2| > 3
        check_pant2 = 1
        Si col_borde1 > [1/3(ancho_ventana) - 5]
            col_hor1_ini = 5
            col_hor1_fin = [1/2(5 + col_borde2)]
            col_hor2_ini = 5
            col_hor2_fin = col_borde2
        Si no
            col_hor1_ini = col_borde1
            col_hor1_fin = [1/2(col_borde1 + col_borde2)]

```

```

        col_hor2_ini = col_borde1
        col_hor2_fin = col_borde2
Si no
    check_pant1 = 1
    Si col_borde1 > [1/3(ancho_ventana) - 5]
        col_hor1_ini = 5
        col_hor1_fin = col_borde2
        col_hor2_ini = 5
        col_hor2_fin = col_borde2
    Si no
        col_hor1_ini = col_borde1
        col_hor1_fin = col_borde2
        col_hor2_ini = col_borde1
        col_hor2_fin = col_borde2

```

De las decisiones anteriores, se observa que el inicio o el fin del borde 1 de horizontal (cuando se cumple ciertas condiciones) es la suma de las columnas de los dos bordes verticales dividido a la mitad, esto se hace dado que para el borde de horizontal punta por la propia forma del zapato, no se hallará dicho borde más de la mitad del zapato, por tanto no hay caso que se realice el proceso del borde a lo largo de todas las columnas entre los dos bordes horizontales. Además hay dos variables (`check_pant1`, `check_pant2`) que se usarán para una el control de un proceso explicado más adelante.

Una vez que se tiene contemplado las posiciones en donde se realizarán los procesos para los dos bordes horizontales como se muestra en la Fig. 3.22, haciendo primeramente el proceso de la primera horizontal y consiguientemente el proceso de la segunda horizontal, pero a diferencia de la figura mencionada, no es necesario hallar el primer borde para pasar al proceso del segundo borde, dado que en varias pruebas en imágenes con alguna dificultad (sombra, posición, borrosidad, etc.) no siempre se puede hallar alguna de estas dos horizontales y se ha notado que la que más dificultad tiene es la horizontal de punta, por tanto también cambia el orden de las horizontales, es decir, la primera horizontal pasa a

ser la horizontal de suela y la segunda horizontal es la horizontal punta, más adelante se explicará una solución para situaciones donde no se hallen las horizontales. De manera general, los procesos de horizontales quedarían de la siguiente manera:

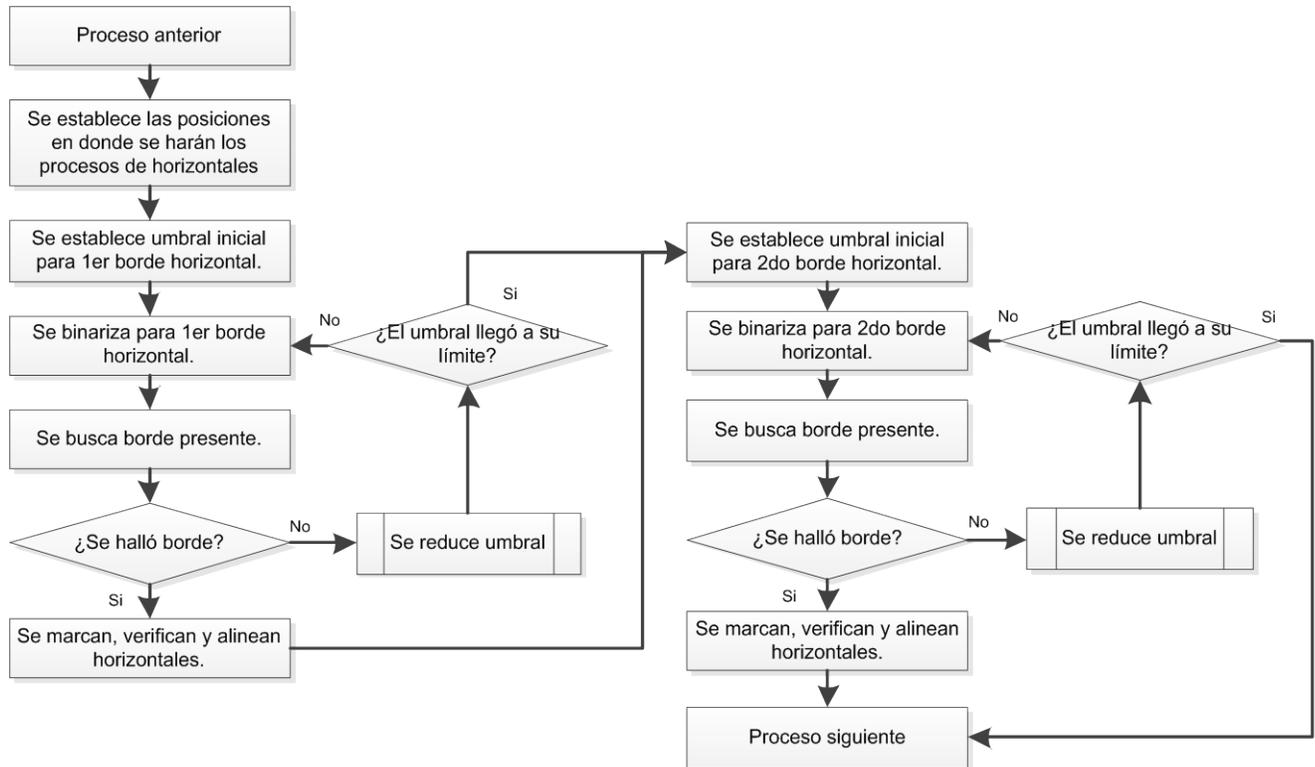


Fig. 3.32: Diagrama para los procesos de las dos horizontales en ventana.

Tal como lo muestra la Fig. 3.32, si alguno o incluso los dos bordes horizontales no son hallados, se pasa al siguiente proceso. Puesto que ya se realizaron los procesos para los 4 bordes, el proceso siguiente es la condición de que si se cumple con todas las características del zapato o no se cumplen.

3.9 Decisión de Presencia de Zapato

Ya realizado todos los procesos para los 4 bordes, elección de zona, ventana, medidas, binarización, búsqueda, reacomodo de ventana, etc. y determinar que la separación entre bordes verticales está en lo correspondiente a un zapato, entonces con una condición simple de la presencia de los 4 bordes se dice que **si es un zapato** y este es el resultado final haciendo que las iteraciones no continúen dado a que son ciclos, se dice que se “rompen” estos ciclos; en el proceso siguiente después de los bordes horizontales simplemente se verifica que se hayan encontrado estos dos bordes dado que los otros dos bordes verticales deben estar presentes ya que no se podrían realizar todos los procesos posteriores, véase la Fig. 3.33.

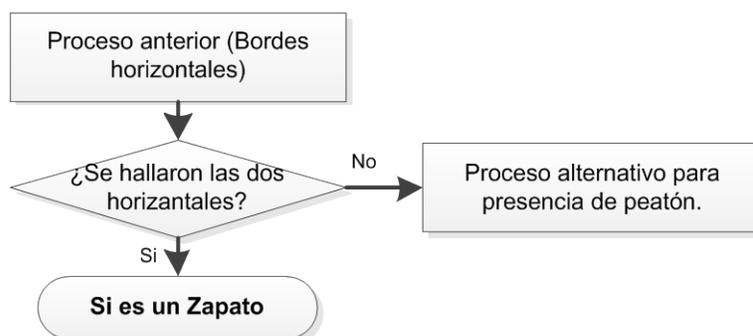


Fig. 3.33: Condición para determinar la presencia de un Zapato.

Cabe resaltar que el algoritmo una vez que encuentra un zapato, interrumpe la búsqueda en el resto de la imagen, por tanto **no se contabilizan el número de zapatos** que pueda haber en la imagen, **la prioridad es hallar el zapato más cercano** ya que es más importante evitar la colisión con el peatón más próximo al vehículo y por tanto, es innecesario saber cuántos zapatos hay en toda la imagen dado que implicaría un gasto en tiempo mayor y no sería viable para un sistema aplicado en línea.

3.10 Superposición de bordes en imagen original

Este proceso solo se realiza para fines demostrativos y mostrar los 4 bordes hallados correspondientes del zapato sobre la imagen, simplemente con sumar el inicio de la posición de fila y columna de la imagen donde la ventana halló el zapato, más la posición de filas columnas de los bordes hallados quedando de la siguiente manera:

```
%% Primer borde vertical
fil_ini_sup_borde1 = fil_ini_borde1 + fil_ini_ventana1
fil_fin_sup_borde1 = fil_fin_borde1 + fil_ini_ventana1
col_sup_borde1 = col_borde1 + col_ini_ventana1

%% segundo borde vertical
fil_ini_sup_borde2 = fil_ini_borde2 + fil_ini_ventana2
fil_fin_sup_borde2= fil_fin_borde2 + fil_ini_ventana2
col_sup_borde2 = col_borde2 + col_ini_ventana2

%% Primer borde horizontal
fil_sup_hor_suela = fil_hor_suela + fil_ini_ventana2
col_ini_sup_hor2 = col_hor2_ini + col_ini_ventana2
col_fin_sup_hor2 = col_hor2_fin + col_ini_ventana2

%% Segundo borde horizontal
fil_sup_hor_punta = fil_hor_punta + fil_ini_ventana2
col_ini_sup_hor1 = col_hor1_ini + col_ini_ventana2
col_fin_sup_hor1 = col_hor1_fin + col_ini_ventana2
```

Para poder plasmar los bordes, es necesario modificar las matrices de la imagen, recordando que una imagen RGB tiene 3 matrices, una para cada color, para una muestra simple se decidió mostrar todos los bordes en color rojo que en RGB es (255, 0, 0), por tanto, para obtener ese color es necesario modificar las 3 matrices

con las posiciones de los bordes ya estipulados en el pseudocódigo anterior quedando de la siguiente forma.

```
%% Primer borde vertical
Para fil_sup1 de fil_ini_sup_borde1 to fil_fin_sup_borde1
    Imagen(fil_sup1, col_sup_borde1, 1) = 255    %% R
    Imagen(fil_sup1, col_sup_borde1, 2) = 0      %% G
    Imagen(fil_sup1, col_sup_borde1, 3) = 0      %% B

%% Segundo borde vertical
Para fil_sup2 de fil_ini_sup_borde2 to fil_fin_sup_borde2
    Imagen(fil_sup2, col_sup_borde2, 1) = 255
    Imagen(fil_sup2, col_sup_borde2, 2) = 0
    Imagen(fil_sup2, col_sup_borde2, 3) = 0

%% Primer borde horizontal
Para col_sup1 de col_ini_sup_hor2 to col_fin_sup_hor2
    Imagen(fil_sip_hor_suela, col_sup1, 1) = 255
    Imagen(fil_sip_hor_suela, col_sup1, 2) = 0
    Imagen(fil_sip_hor_suela, col_sup1, 3) = 0

%% Segundo borde horizontal
Para col_sup2 de col_ini_sup_hor1 to col_fin_sup_hor1
    Imagen(fil_sip_hor_punta, col_sup2, 1) = 255
    Imagen(fil_sip_hor_punta, col_sup2, 2) = 0
    Imagen(fil_sip_hor_punta, col_sup2, 3) = 0
```

Con lo anterior entonces ya es posible mostrar los 4 bordes hallados sobre la imagen original y visualizar de esta forma si realmente los bordes quedaron posicionados donde se encuentra el zapato en la imagen mostrado en la Fig. 3.34.



Fig. 3.34: Imagen final con los 4 bordes hallados superpuestos.

Tal como se ha mencionado anteriormente, hay casos donde alguna o ninguna de las dos horizontales son halladas, para la imagen final, por tanto, solo se mostrarán aquellos bordes hallados incluso solamente los dos bordes verticales continuando con el proceso alternativo para indicar la presencia de un peatón, tal como se muestra en la Fig. 3.35.



Fig. 3.35: Imagen final mostrando los bordes hallados.

3.11 Proceso alternativo subsecuente para presencia de un peatón

Cuando no se halla alguno o ninguno de los dos bordes horizontales, por obviedad no se cumple la condición para determinar que si hay un zapato en la imagen, pero dado que si hay dos bordes verticales con la separación suficiente para un zapato, se realiza un proceso alternativo para averiguar si hay la presencia de un peatón.

Este proceso consiste en la detección de una porción de la pierna cerca de la ubicación donde se hallaron los bordes incompletos del zapato, consistirá en hallar dos bordes verticales largas que no sobrepasen cierta separación que, al estudiar varias muestras, la separación debe ser al menos un tercio de la separación entre las dos verticales del zapato, tal como se observa en la Fig. 3.36.

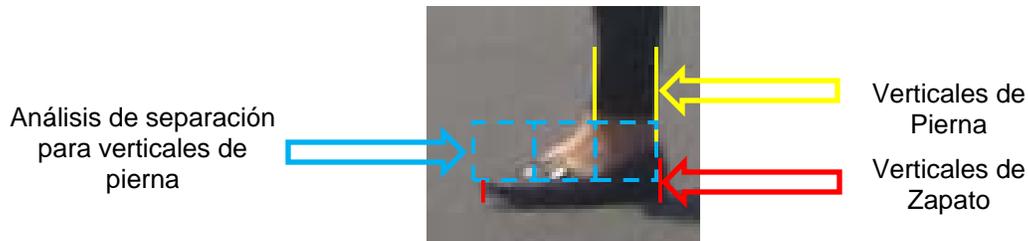


Fig. 3.36. Establecimiento de la separación para pierna.

Para reposicionar la ventana lo suficiente para que abarque la sección de la pierna que es continua al zapato, se tomará como referencia la fila de la primera posición del borde correspondiente al talón, es decir, al borde hallado que está más en la parte superior de la ventana, además con ello es posible buscar la prolongación de la vertical del talón dado a la propia forma de la pierna en estado vertical, por tanto la reposición de la ventana sería:

```
%% Filas
fil_fin_vent_pant = fil_fin_ventana2 - alto_ventana - fil_ini_borde2
fil_ini_vent_pant = fil_fin_vent_pant - alto_ventana + 1

%% Columnas
col_ini_vent_pant = col_ini_ventana2
col_fin_vent_pant = col_fin_ventana2
```

Reubicada la ventana se procede por tanto a los proceso para determinar si hay dos bordes verticales correspondientes a una porción de pierna, se estipuló que para ambas verticales deben tener un mínimo de 5 pixeles seguidos cada una para considerarlas un borde de pierna, esto es puesto que se especula que

cuando se hallan bordes de zapato la pierna está en una posición recta y vertical porque la suela del zapato se asume está paralela al pavimento como se observa en la Fig. 3.37.



Fig. 3.37: Reposicionamiento de ventana para detección de pierna.

Los procesos son similares a los de los bordes verticales, es decir, binarización con umbral adaptable y limitado y búsqueda de bordes en la imagen binarizada, salvo a que ya no se hace un proceso de líneas vecinas (dado que encontrando 5 píxeles seguidos ya se considera borde) y no hay superposición de la pierna, ya que lo que interesa es hallar solamente indicios de pierna.

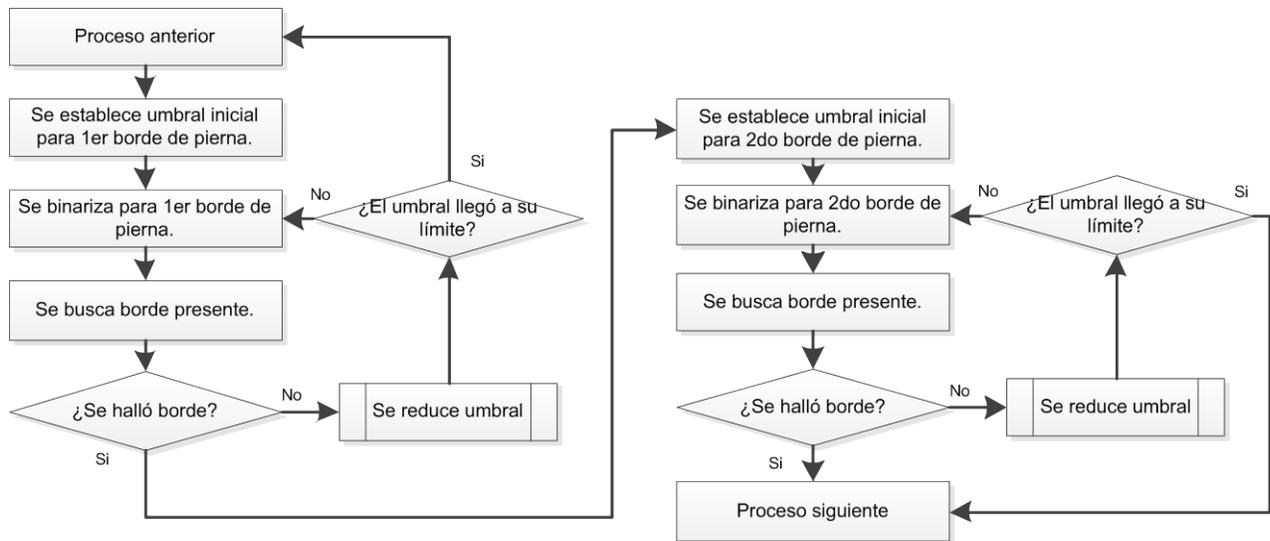


Fig. 3.38: Diagrama de los procesos para los bordes de pierna.

Tal como se muestra en la Fig. 3.38, es necesario hallar al menos un borde para indicar que hay una porción de pierna, esto es dado a que se hallaron casos donde no es posible detectar los dos bordes, pero el borde hallado debe estar entre los límites de la separación establecida. Recordando las variables `check_pant1` y `check_pant2`, éstas servirán para indicar si el borde hallado pertenece a un lado cercano al talón o el otro extremo de la pierna.

Por tanto, si se hallan los dos bordes y comparando la separación entre ambos que debe estar en los límites establecidos: un tercio de la separación de los dos bordes verticales del zapato, si se cumple la condición entonces se dice que **hay presencia de zapato y pierna** y se “rompen” las iteraciones de la ventana.

```
Si (col_pant1 - col_pant2) >= sep_pant
    "Hay presencia de zapato y pierna"
    Romper iteraciones iter1 e iter2
Si no
    "No corresponde a pierna"
    Seguir iteraciones iter1 e iter2
```

Para cuando se halle un solo borde de la pierna, primeramente para el proceso del primer borde se realiza en toda la sección establecida por los dos bordes verticales del zapato, es decir, se realiza dentro de toda el área limitada por dichos bordes, por ende si no se le hallase se pasa a un proceso anterior que en sí es posicionar la ventana en la siguiente iteración para comenzar de nuevo todos los procesos desde el principio.

Si se halla algún borde se activa una variable que es la que da inicio al proceso del segundo borde, entonces cuando se activa la variable `check_borde1`, indica que el borde de talón está de izquierda a derecha y debe estar un borde con una posición establecida, ya sea muy cerca del borde de talón de zapato o si es del

otro borde, que sea mayor a la separación establecida, si se cumple alguna condición entonces se dice que **hay presencia de vertical izquierda(talón)-derecha(extremo) correspondiente a pierna** y se rompen las iteraciones, si no se cumple entonces se dice que **no hay indicios de pierna** y se siguen las iteraciones para la ventana.

Cuando se activa `check_borde2` el talón ahora es de derecha a izquierda e igual se examina que el borde esté muy cerca del talón que ahora está en el extremo derecho, o sea menor a la separación multiplicada por 2, esto dado el sentido propio del talón puesto que la separación es un valor en columnas y éstas siempre se leen de izquierda a derecha, por tanto, al estar el talón a la derecha se tiene que medir la separación “al revés”, es decir, un tercio después del talón, que en sentido normal son dos tercios para que se establezca como el borde del otro extremo de la pierna, si se cumple alguna condición entonces se dice que **hay presencia de vertical derecha(talón)-izquierda(extremo) correspondiente a pierna**, si no se cumplen entonces se dice que **no hay indicios de pierna** y las iteraciones siguen normalmente en caso de faltar completarlas para toda la imagen.

```
Si check_borde1 = 1
    Si col_pant1 < (col_borde1 + 5)
        “Hay presencia de vertical izquierda correspondiente a pierna”
        Romper iteraciones iter1 e iter2
    Si no, si col_pant1 > sep_pant
        “Hay presencia de vertical derecha correspondiente a pierna”
        Romper iteraciones iter1 e iter2
    Si no
        “No hay indicios de pierna”
        Seguir iteraciones iter1 e iter2
Si no, si check_borde2 = 1
    Si col_pant2 > (col_borde2 - 5)
        “Hay presencia de vertical derecha correspondiente a pierna”
        Romper iteraciones iter1 e iter2
```

```
Si no, si col_pant2 < (sep_pant*2)
    "Hay presencia de vertical izquierda correspondiente a pierna"
    Romper iteraciones iter1 e iter2
Si no
    "No hay indicios de pierna"
    Seguir iteraciones iter1 e iter2
```

Así con lo anterior, se tiene completo todo el algoritmo para la detección de zapatos en imágenes para todas las zonas y para distintos tipos de zapatos junto con las dificultades que se puedan presentar para su detección. El algoritmo completo se muestra en la Fig. 3.39, Fig. 3.40 y Fig. 3.41.

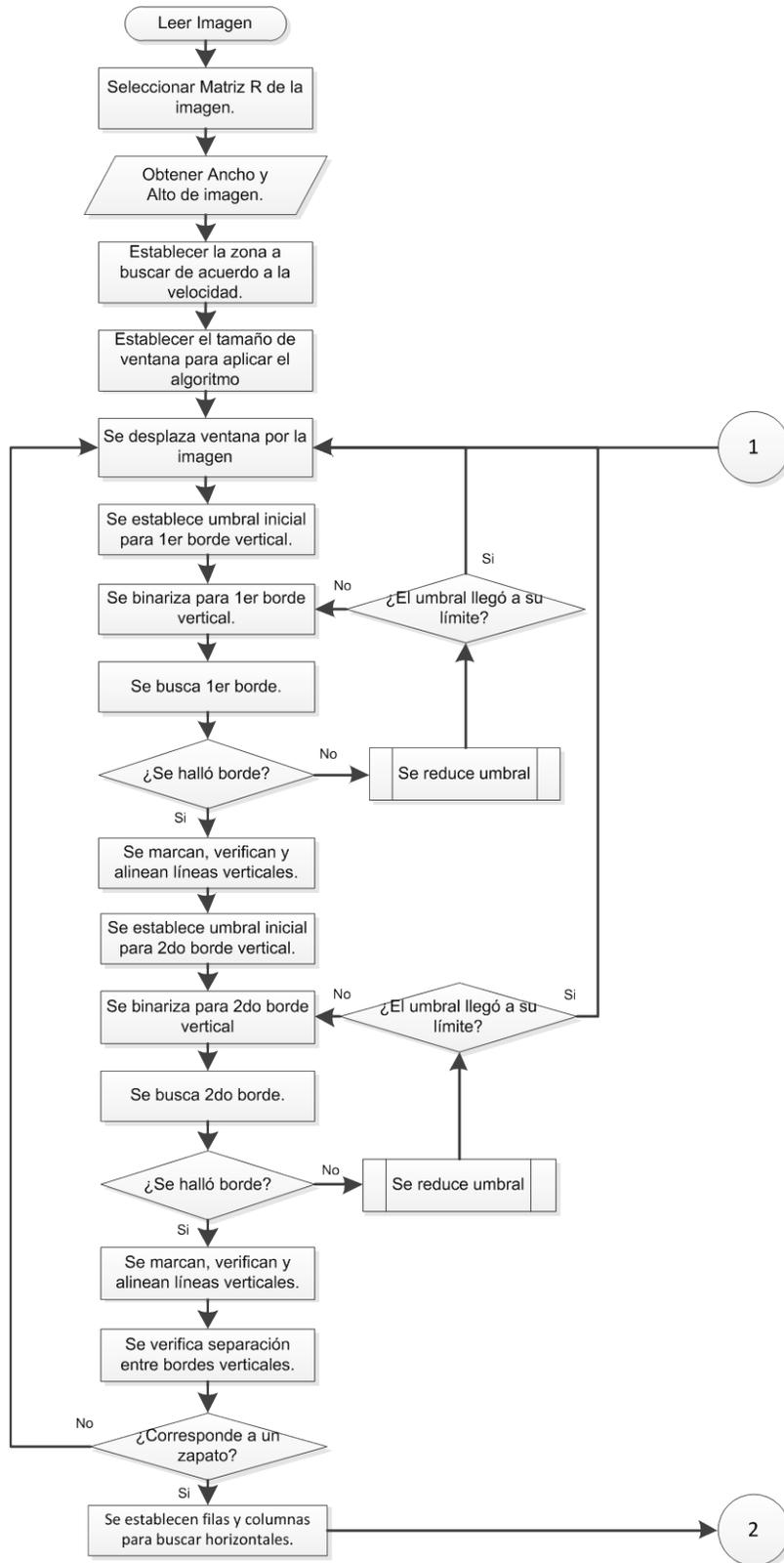


Fig. 3.39: Primera parte del Algoritmo Completo.

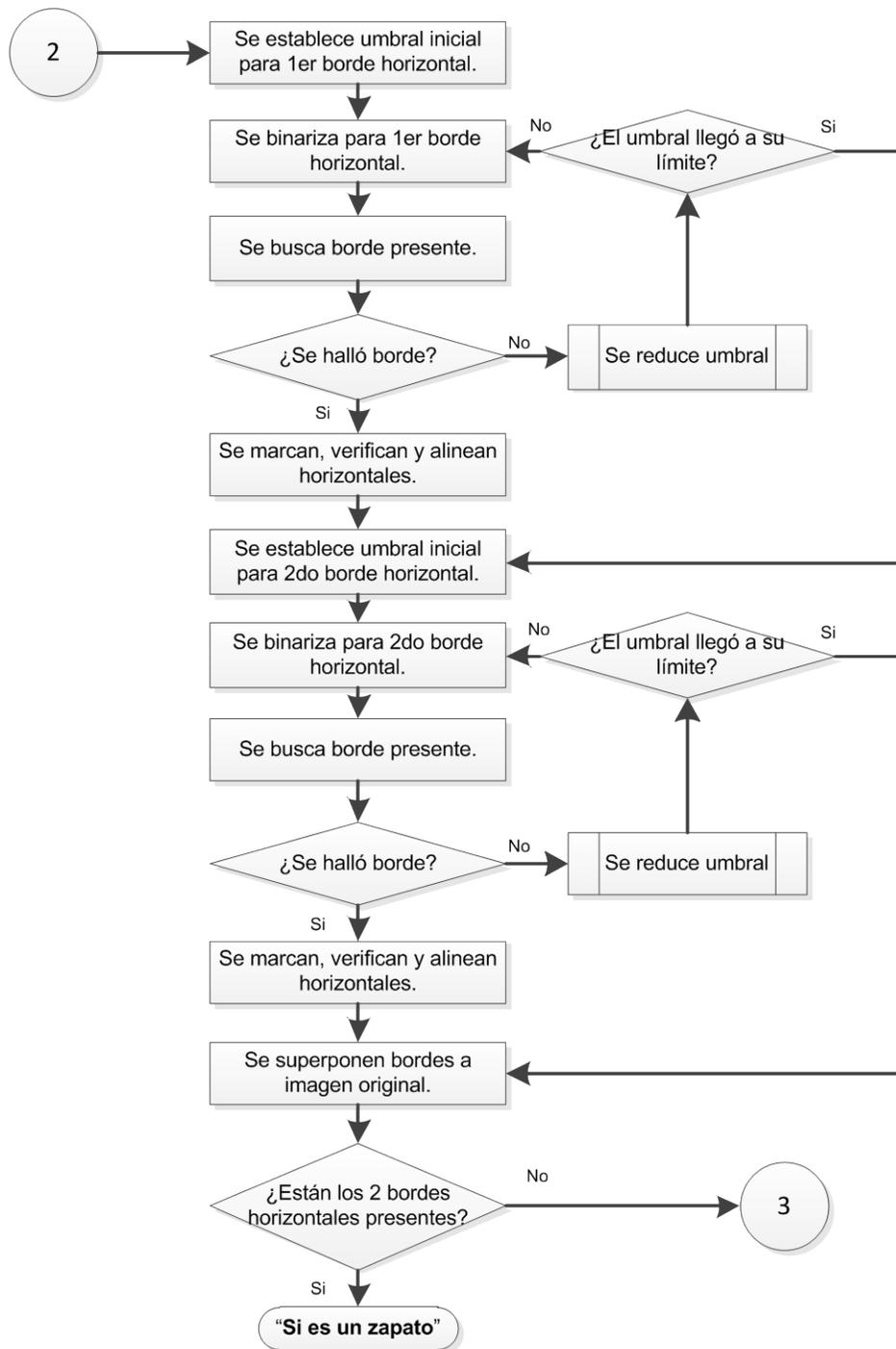


Fig. 3.40: Segunda parte del Algoritmo Completo.

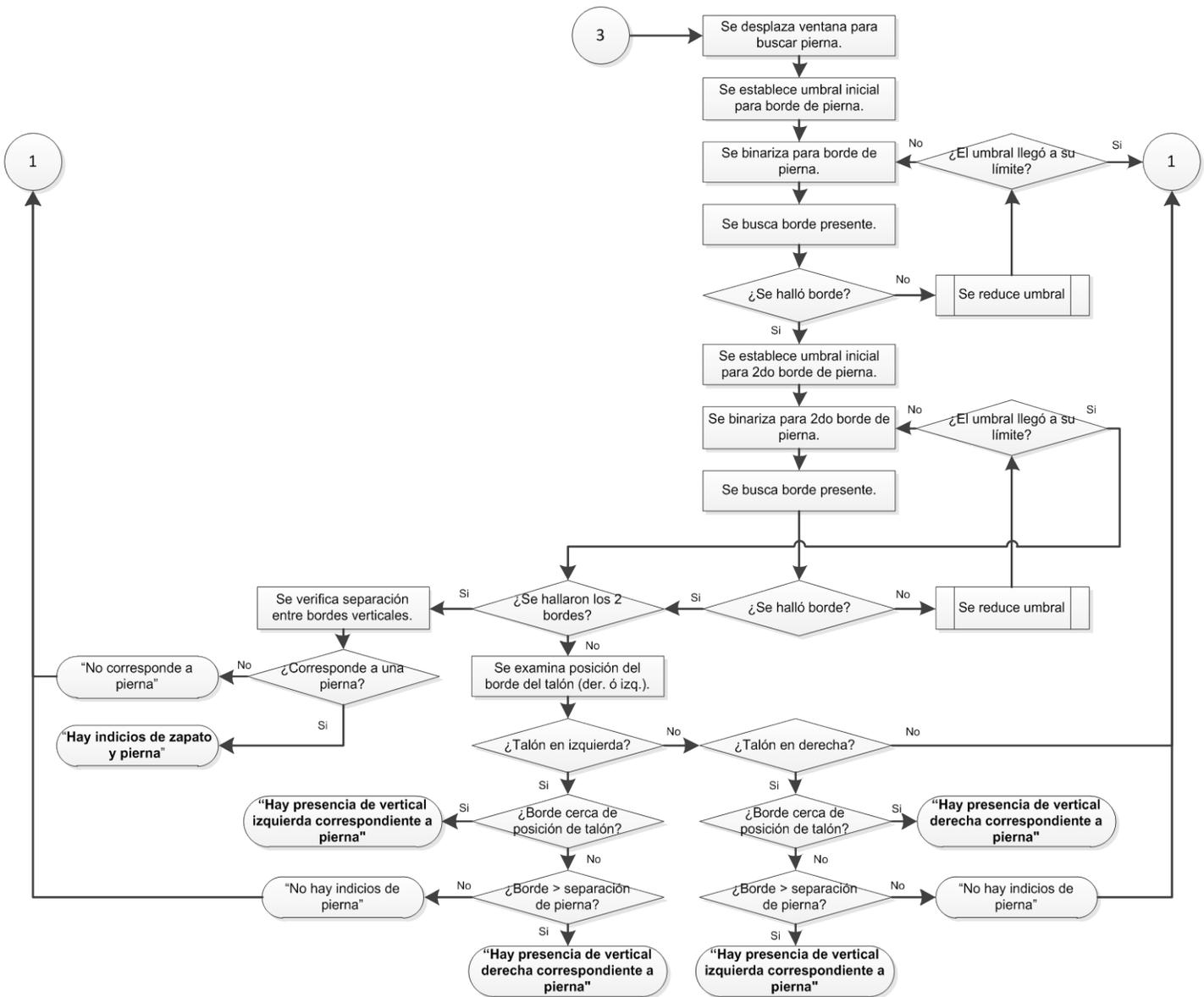


Fig. 3.41: Tercera parte del Algoritmo Completo.

Al tener completo el algoritmo se procede a su aplicación en imágenes para observar su funcionamiento y discutir sus resultados, esto se detalla en el siguiente capítulo.

Capítulo 4

RESULTADOS Y DISCUSIÓN

Para el desarrollo y las pruebas del algoritmo, se dispuso de una computadora portátil con las características mostradas en la Tabla 4.1:

Tabla 4.1: Características del equipo utilizado para el desarrollo y prueba del algoritmo.

Hardware	Características
Sony Vaio FZ250FE	Procesador: Core 2 Duo 2.0 GHz Memoria RAM: 4 GB Sistema Operativo: Windows 7 64 bits.
Software	
Matlab	Versión: R2010a (7.10.0) 64 bits.

Las imágenes tomadas tienen una resolución de 6 Megapíxeles (2816 x 2112) provenientes de la cámara marca Canon en formato jpg, se hicieron dos recorridos para la toma de imágenes en días diferentes, el primer recorrido fue en el centro de la ciudad de Puebla y el segundo fue en el centro y una plaza comercial conocida de la ciudad de Cholula, Puebla.

4.1 Resultados en imágenes con zapatos individuales

Se muestran algunos resultados aplicando el algoritmo para distintos zapatos en la Tabla 4.2, en algunos se voltearon verticalmente para observar la efectividad del algoritmo puesto que debe ser indistinto que si el primer borde que encuentra es talón o punta para continuar con el siguiente proceso.

Tabla 4.2: Resultados de imágenes de zapatos.

Imagen Original	Imagen Procesada
	
	
	
	
	
	
	
	
	
	

4.2 Resultados con el algoritmo completo

Recordando que hay 5 zonas que dependen de la velocidad, para todas las imágenes tomadas en los recorridos, se buscaron peatones presentes y se conforme a la zona se seccionaron manualmente las imágenes donde había cabida en la zona, se hizo así porque las tomas con la cámara se hizo se forma manual y sin soporte, es decir, solamente sosteniendo la cámara con la mano además que el campo de vista es más amplio conforme a la zona de interés, lo que sería similar a que la imagen a procesar fuera una imagen completa de menor resolución puesto que son menos pixeles para la misma área.



Fig. 4.1: Comparación de dos imágenes para una misma zona.

Esta particularidad hace que es posible aplicar el sistema con una cámara a diferentes resoluciones, por ende que entre mayor resolución mayor detalle hay en la imagen y por tanto mayor distancia para detección se tendrá.

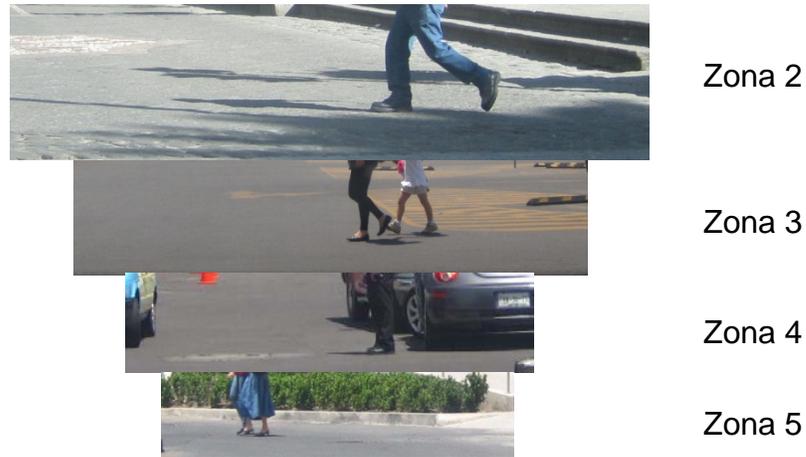


Fig. 4.2: Ejemplos de secciones de imágenes para distintas zonas.

De la Fig. 4.2, se observan muestras de las zonas 2 a la 5 en diferentes lugares, notando también que el pavimento es distinto y además se presentan dificultades como sombras tal como se aprecia en la imagen de la zona 2.

Aplicando el algoritmo a todas las imágenes seccionadas, en total se pusieron a prueba 46 imágenes para todas las zonas, donde 3 corresponden a la zona 2, 7 a la zona 3, 18 a la zona 4 y 18 a la zona 5, se encontraron más casos en zonas lejanas pues es donde se hallaron más peatones además que observando la Fig. 4.1 el tablero obstaculizó en varios casos para las zonas más cercanas, los resultados obtenidos se dividieron en 3 categorías Positivos: donde se da una correcta detección, Negativos: donde no hay detección del zapato y Falsos Positivos: donde hay detección pero no correspondiente a un zapato en la imagen.

Tabla 4.3: Resultados de las distintas imágenes aplicadas al algoritmo.

Zona	Número de imágenes por zona		
	Positivos	Negativos	Falsos Positivos
2	2	1	
3	6	1	
4	17	1	
5	16	1	1
Total:	41	4	1

De la Tabla 4.3, se observa que hay muy buenos resultados para la detección, dado que de las 46 imágenes, 41 fueron detecciones correctas que es un 89% de efectividad, las siguientes figuras muestran ejemplos de resultados para los tres casos.

4.2.1 Positivos

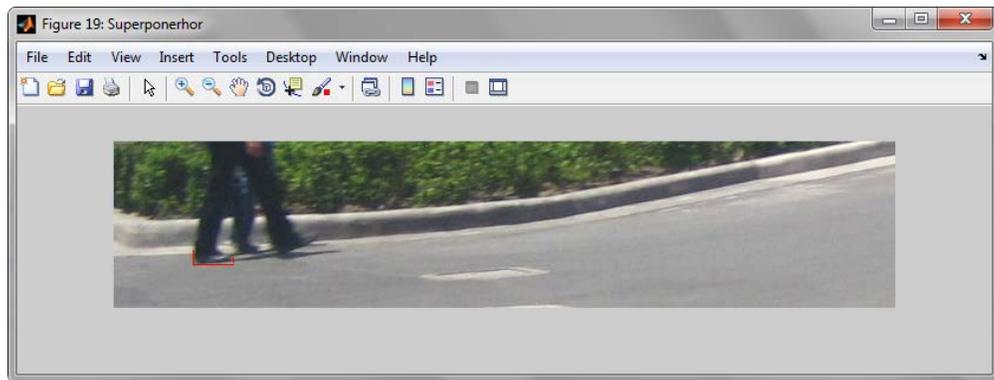


Fig. 4.3: Detección correcta con proceso de pierna en zona 4.



Fig. 4.4: Detección correcta en zona 5.

En la Fig. 4.3 y Fig. 4.4, se muestran los casos de detección tanto para un zapato completo (Fig. 4.4) como la detección de características de una pierna (Fig. 4.3) en zonas diferentes. Se recuerda que solo se muestra el primer zapato sin contabilizar los que hay en toda la imagen como se explica en la sección 3.9 del capítulo 3.

4.2.2 Negativos

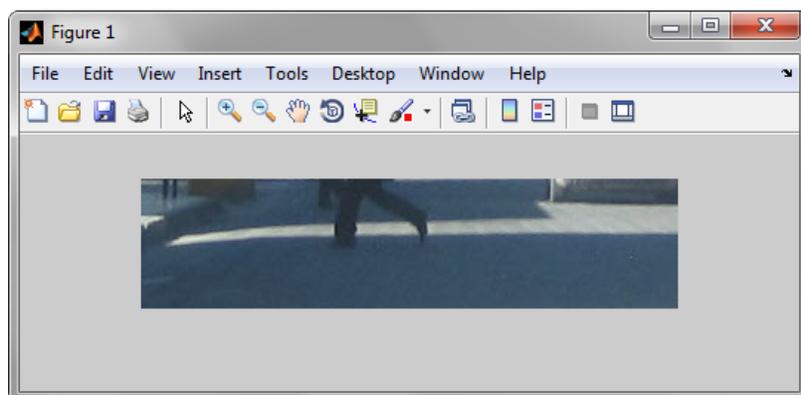


Fig. 4.5: No detección en zona 5.

En la Fig. 4.5 se muestra un caso en donde no hubo detección alguna, puede deberse al mínimo contraste que tiene el zapato con el pavimento oscurecido por la sombra.

4.2.3 Falsos Positivos

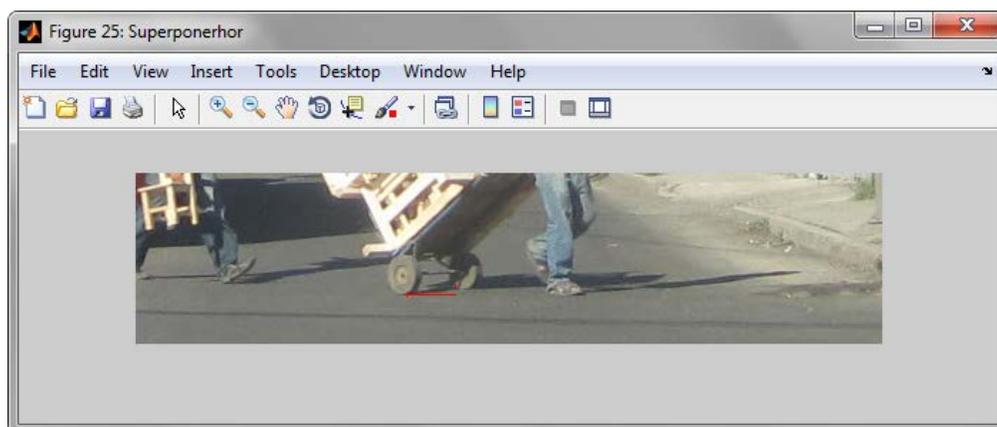


Fig. 4.6: Detección no correspondiente a un zapato en zona 5.

La Fig. 4.6 es el único caso de falso positivo que se presentó, como se observa la detección no corresponde conforme al zapato del peatón sino a un objeto presente en la imagen. Este caso debido a que tan pronto hubo una detección el sistema se detuvo pero si hubiera seguido buscando habría grandes posibilidades que el

sistema detectara a la persona, pero en particular ayuda a que se detectó un obstáculo que no es zapato dado a que esta detección se considera un riesgo por el propio obstáculo. En las muestras tomadas no hubo el caso de detecciones sin obstáculo.

4.2.4 Prueba en asfaltos

Para verificar que no hay presencia de zapato sobre una imagen donde solo esté el asfalto, se presentan dos pruebas con dos tipos de asfalto mostrados en la Fig. 4.7 y en ambos casos el algoritmo no indicó presencia de zapato.



Fig. 4.7: Asfaltos probados en el algoritmo.

4.3 Discusión de Resultados

De la Tabla 4.3, al tomar el total de resultados en porcentajes, se tiene que un 89% corresponde a Positivos, 9% a Negativos y un 2% a Falsos Positivos, lo cual indica que el algoritmo resulta eficiente para la detección de peatones en imágenes de un entorno de vialidad de la ciudad.

Si tomamos en cuenta que además el algoritmo no se aplica a una imagen completa dado a las zonas de interés establecidas y además que es aplicada por ventanas se reduce la relación en cuanto a aplicación, por ejemplo, una imagen de la zona 1 tiene un ancho de 2816 por alto aproximado de 700 pixeles, la ventana correspondiente a la zona es de 280 x 140, si se divide el alto y el ancho de la imagen y la ventana se tiene una relación de aproximada de 10 x 5, es decir, se reduce considerablemente las regiones de interés para toda la imagen, que serían un total de 50 lo que reduce el tiempo computacional para el procesamiento.

En cuanto al tiempo de ejecución para cada imagen, por la propiedad que son de diferente tamaño conforme a la zona y dependiendo de dónde se localice el zapato en la imagen, el tiempo varía mucho para cada imagen, en la Tabla 4.4 se muestran los tiempo relativos de ejecución de algunas imágenes procesadas.

Tabla 4.4: Tiempo relativo de ejecución para diversas imágenes procesadas.

Prueba	Tiempo relativo	Prueba	Tiempo
zona 2-1	1.9	t2zona 4 - 6	1.3
zona 2-2	3.2	t2zona 4 - 7	1.9
zona 3-1	1.9	t2zona 4 - 8	1.9
zona 3-2	1.8	t2zona 4 - 9	1.15
zona 3-3.	1.8	t2zona 4 - 10	1.6
zona 4-1	2.6	t2zona 5 - 1	2.1
zona 4-2	1.6	t2zona 5 - 2	1.7
zona 4-3	1.3	t2zona 5 - 4	1.7
zona 5-1	1.8	t2zona 5 - 5	1.0

Recordando que el algoritmo se implementó en formato “m-code” de Matlab y en comparación contra un código compilado, el tiempo de ejecución se reduce al menos en un factor de 5, si se emplea código orientado a hardware como VHDL, se reduce aún más el tiempo de ejecución por lo que lo hace un buen candidato a implementación en línea.

Capítulo 5

CONCLUSIONES Y TRABAJO FUTURO

5.1 Conclusiones

La detección de peatones en base a la detección de zapatos es un área prometedora, dado los resultados del algoritmo presentado para la detección de las características comunes de los zapatos como lo son talón, punta y suela, dependiendo de la calidad de la imagen, si no se encuentra alguna característica el algoritmo es capaz de buscar características de la pierna donde hay presencia del zapato, tales características son que la pierna forma dos verticales casi paralelas.

El algoritmo desarrollado es en base a la detección de bordes por ventaneo con operaciones no complejas que recorre la imagen para buscar las características mencionadas, esto hace que se reduce el costo computacional. Se reduce el orden del número de iteraciones.

La búsqueda de zapatos es una nueva forma para identificar peatones en una imagen, en ningún otro trabajo se menciona o se ha aplicado la búsqueda del

peatón por sus zapatos, tras analizar varias imágenes se determinó que los zapatos son lo que menos se obstaculizan por algún objeto que lleve la persona o por la posición que tenga ésta misma.

Es un método alternativo tomando en cuenta que solo se tiene un sensor (la cámara fotográfica) debido a que los detectores comerciales tienen más de dos tipos de sensores y su detección es limitada a ciertas restricciones.

5.2 Trabajo Futuro

Hacer más pruebas con imágenes en otros entornos reales de tráfico de distintas partes de la ciudad a diferente horario para incluir mayor variedad de peatones y si es posible, en diferentes entornos climatológicos como lluvia y niebla para poder incluir características que ayuden a la detección en estos entornos.

Corroborar que la posición de la cámara no afecta al sistema, pero se recomienda hacer un estudio del posicionamiento de la cámara fotográfica en un vehículo para poder tener el mejor ángulo de visión para todas las zonas y no se vea afectada por factores como polvo, agua, arena, etc. sin que además obstaculice al conductor.

Probar el algoritmo en un código orientado a hardware como VHDL, así para poder implementarlo en tarjetas de desarrollo FPGA y analizar el tiempo de ejecución y observar la factibilidad de un prototipo en tiempo real.

BIBLIOGRAFÍA

- [1] INSP, *México, sede del Secretariado de la Red de Investigación sobre Lesiones causadas por el Tránsito*, Cuernavaca, Morelos: Boletín de Prensa, 2010.
- [2] INSP, «Instituto Nacional de Salud Pública,» 10 diciembre 2010. [En línea]. Available: <http://www.insp.mx/noticias/entrevistas-de-radio/1487-consecuencias-de-los-accidentes-viales-en-los-jovenes.html>. [Último acceso: 5 octubre 2011].
- [3] C. Hilario Gómez, *Detección de Peatones en el Espectro Visible e Infrarrojo para un Sistema Avanzado de Asistencia a la Conducción*, Madrid: Universidad Carlos III de Madrid, 2008.
- [4] D. Gerónimo, A. M. López y A. D. Sappa, *Survey of Pedestrian Detection for Advanced Driver Assistance Systems*, vol. 32, I. C. Society, Ed., IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 2010.
- [5] AUDI USA, «AUDI A6,» 2011. [En línea]. Available: http://www.audiusa.com/us/brand/en/models/a6_sedan/explore/performance.html#source=http://www.audiusa.com/us/brand/en/models/a6_sedan/explore/technology_safety.html&container=page. [Último acceso: 5 octubre 2011].
- [6] AUDI USA, «AUDI A7,» 2011. [En línea]. Available: http://www.audiusa.com/us/brand/en/models/a7/explore/exterior_design.html#source=http://www.audiusa.com/us/brand/en/models/a7/explore/technology_safety.html&container=page. [Último acceso: 5 octubre 2011].
- [7] AUDI USA, «AUDI A8,» 2011. [En línea]. Available: http://www.audiusa.com/us/brand/en/models/a8/explore/performance_handling.html#source=http://www.audiusa.com/us/brand/en/models/a8/explore/technology_safety.html&container=page. [Último acceso: 5 octubre 2011].

- [8] BMW USA, «BMW Series 5,» 2011. [En línea]. Available: http://www.bmwusa.com/standard/content/vehicles/2012/5/5explore.aspx#safety/night_vision. [Último acceso: 5 octubre 2011].
- [9] BMW USA, «BMW Series 7,» 2011. [En línea]. Available: <http://www.bmwusa.com/Standard/Content/Vehicles/2011/7/allnew7series.aspx>. [Último acceso: 5 octubre 2011].
- [10] MERCEDES-BENZ, «MB Class S,» 2011. [En línea]. Available: <http://www.mbusa.com/mercedes/vehicles/models/class-S/bodystyle-SDN>. [Último acceso: 5 octubre 2011].
- [11] MERCEDES-BENZ, «MB Class CL,» 2011. [En línea]. Available: <http://www.mbusa.com/mercedes/vehicles/models/class-CL/bodystyle-CPE>. [Último acceso: 5 octubre 2011].
- [12] MERCEDES-BENZ, «MB Class CLS,» 2011. [En línea]. Available: <http://www.mbusa.com/mercedes/vehicles/model/class-CLS/model-CLS550C>. [Último acceso: 5 octubre 2011].
- [13] VOLVO CARS, «VOLVO XC60,» 2011. [En línea]. Available: <http://www.volvocars.com/us/all-cars/volvo-xc60/details/pages/features.aspx#/?category=2dd7a937-46d5-4251-8015-12939a77a480>. [Último acceso: 5 octubre 2011].
- [14] VOLVO CARS, «VOLVO XC70,» 2011. [En línea]. Available: <http://www.volvocars.com/us/all-cars/volvo-xc70/details/pages/features.aspx#/?category=2dd7a937-46d5-4251-8015-12939a77a480>. [Último acceso: 5 octubre 2011].
- [15] VOLVO CARS, «VOLVO S60,» 2011. [En línea]. Available: <http://www.volvocars.com/us/all-cars/volvo-s60/details/pages/features.aspx#/?category=2dd7a937-46d5-4251-8015-12939a77a480>. [Último acceso: 5 octubre 2011].
- [16] VOLVO CARS, «VOLVO S80,» 2011. [En línea]. Available: <http://www.volvocars.com/us/all-cars/volvo-s80/details/pages/features.aspx#/?category=2dd7a937-46d5-4251-8015-12939a77a480>. [Último acceso: 5 octubre 2011].
- [17] C. Papageorgiou y T. Poggio, *Trainable Pedestrian Detection*, Cambridge, MA. USA: Center for Biological and Computational Learning Artificial Intelligence Laboratory MIT, 2000.

- [18] D. M. Gavrila, *Pedestrian Detection from a Moving Vehicle*, Wilhelm Runge St. Germany: Image Understanding Systems, DaimlerChrysler Research, 2000.
- [19] A. Shashua, Y. Gdalyahu y G. Hayun, *Pedestrian Detection for Driving Assistance Systems: Single-frame Classification and System Level Performance*, Parma, Italy: Proc. IEEE Intelligent Vehicles Symp., 2004.
- [20] L. Zhao y C. E. Thorpe, *Stereo- and Neural Network-Based Pedestrian Detection*, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, 2000.
- [21] A. Broggi, M. Bertozzi, A. Fascioli y M. Sechi, *Shape-based Pedestrian Detection*, Dearbon (MI), USA: IEEE Intelligent Vehicles Symposium, 2000.
- [22] D. M. Gavrila, *A Bayesian, Exemplar-Based Approach to Hierarchical Shape Matching*, Amsterdam, The Netherlands.: IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 2007.
- [23] C. Hilario, J. M. Collado, J. M. Armingol y A. de la Escalera, *DETECCIÓN DE PEATONES PARA VEHÍCULOS INTELIGENTES BASADA EN MODELOS DE CONTORNOS ACTIVOS Y VISIÓN ESTÉREO*, Madrid, España: Escuela Politécnica Superior, Univ. Carlos III de Madrid., 2004.
- [24] E. Jiménez Camacho, *Medición de distancias por medio de procesamiento de imágenes y triangulación, haciendo uso de cámaras de video*, Puebla, México: Departamento de Computación, Electrónica y Mecatrónica, UDLAP, 2009.
- [25] SCT, *Manual de Proyecto Geométrico de Carreteras*, 4ta ed., México: Secretaría de Comunicaciones y Transportes, 1991, p. 17.
- [26] Dirección General de Tráfico, *Manual Experimental de Prevención de Riesgos Laborales (accidentes de trabajo-accidentes de tráfico)*, Salamanca, Esp.: Gráficas Varona S.A., 1999, pp. 81-84.
- [27] fleet safety forum, *Keeping your distance – A Fleet Safety Forum information sheet for fleet managers and drivers*, Brake: the road safety charity, 2008.
- [28] «Automotores Online,» 2011. [En línea]. Available: <http://automotoresonline.blogspot.com>. [Último acceso: 5 octubre 2011].

- [29] M. Steindel y J. Montufar, *Automated Pedestrian Detection for Safety and Accessibility*, Whistler, British Columbia: 18th Canadian Multidisciplinary Road Safety Conference, 2008.
- [30] L. Bañón Blázquez, *Manual de Carreteras*, Universidad de Alicante, Escuela Politécnica Superior, 1999.
- [31] Universidad Politécnica de Catalunya, 2011. [En línea]. Available: <http://upcommons.upc.edu/pfc/bitstream/2099.1/3316/6/55865-6.pdf>. [Último acceso: 5 octubre 2011].
- [32] J. Kovac, P. Peer y F. Solina, *Human Skin Colour Clustering for Face Detection*, Ljubljana, Slovenia: University of Ljubljana, 2003.